I/O NEWS

ProCall/PC Plus
RDISK
Software Resource Guide

THE OFFICIAL PUBLICATION OF THE INTERNATIONAL ASSOCIATION OF CROMEMCO USERS

Volume Five, Number Two

Single Copy Price $10.00

TELECOMMUNICATIONS New Software Links Cromemco to Mainframes

As computers proliferate, communications among computers become increasingly important. This communication may take the form of clustered PC's (communicating to a Cromemco, say , using ProCall or PC Works software), local area networks interconnecting computers withing a facility, or telecommunications links connecting computers in one facility to a distant computer (say, a mainframe). Of these, the telecommunications link presents the biggest technical challenge. The reason for this is the relatively limited bandwidth available over most physical links (such as telephone lines) and the possibility of the corruption of the data being sent due to electrical noise (caused by lightning, crosstalk, or other physical effects). A number of telecommunications protocols have been developed in order to maximize the use of available bandwidth while at the same time providing error correction in the event of signal corruption. Two of the most important of these protocols, 3270 bisync and X.25, are now supported by Cromemco under both the Cromix and UNIX operating systems.

Both the 3270 and X.25 standards use what is called synchronous communication. Synchronous communication differs from the more familiar asyn-

Continued on page 12

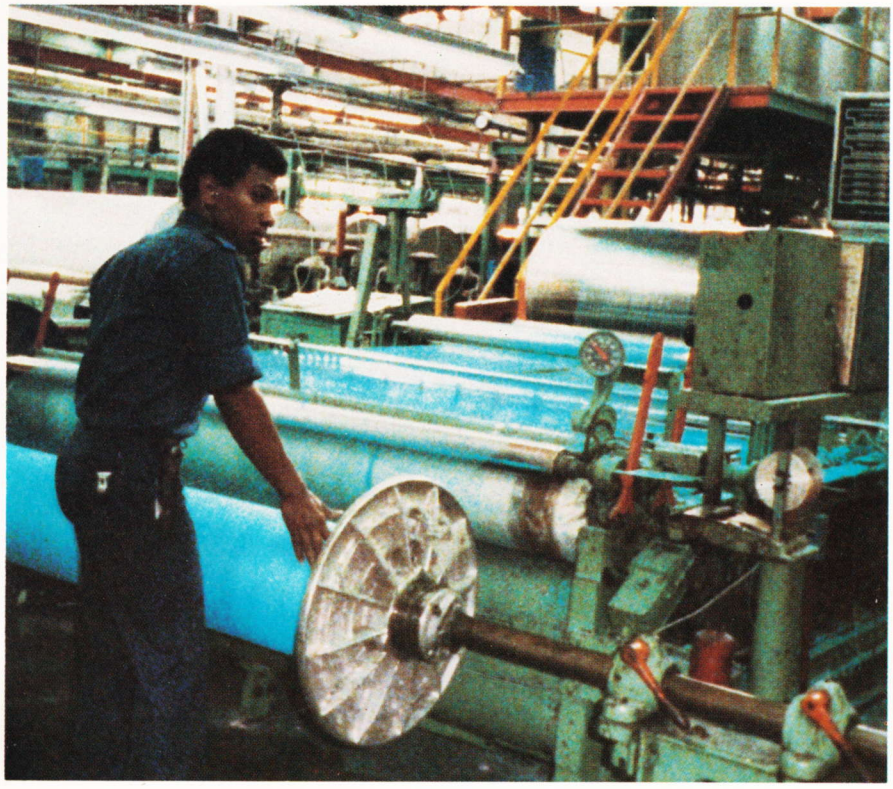
dBIII Compiler for Cromix-Plus

by Rich Mitchell

Many business, scientific, engineering, and military users have used dBASE II on Cromemco systems for years. Although pleased with the ability to use such a tool, they have hungered for 68000 speed, dBASE III features, and a better interface to the Cromix operating system. With the release of **dBIII Compiler for Cromix-Plus**, Cromemco users can now have what they desire.

dBIII Compiler for Cromix-Plus supports the non-interactive features of dBASE III, Cromix-Plus operating system interface including file and record locking for multi-user protection, data file compatibility with dBASE II and dBASE II, and, of course, runs com-

Continued on page 13



Pantex textile factory in Medellin, Colombia.

Cromemco in South America

South America is a land of contrasts, from impenetrable Amazon jungle to open deserts, from coastal ranges to the towering Andes. The economic development and industrialization of the countries of South America is also a landscape of contrasts. Picture this: old town Quito, a sprawling open market woven along the narrow, hilly streets of the city. Merchants selling every description of produce and wares just as they have done for generations. It is as if Safeway and Sears together disgorged their contents and consigned them to the street merchants of San Francisco. Now step into an open doorway and see lines of people standing at bank-like windows applying for consumer credit. Yes, with a Multiplan Card from Ruben Herrmann goods can be bought on credit. And the Multiplan system, providing consumer credit to over 27,000 people, is run on the latest technology Cromemco System 100.

This System 100, which was installed by **Compusystems** — Cromemco's distributor in Quito — is just one exam-

Continued on page 13

table of contents

January/February, 1986 Volume Five, Number Two

I/O NEWS *The Official Publication of The International Association of Cromemco Users is available through membership in the association. Editorial and advertising policies are designed for the enlightenment of the members in regard to new uses for, and developments of, Cromemco products and other products compatible with Cromemco systems.*

COVER FEATURES

ARTICLES & FEATURES

DEPARTMENTS

I/O NEWS [ISSN 0274-9998] is published bi-monthly by The International Association of Cromemco Users (a California corporation), P.O. Box 17658, Irvine, CA 92713. General Offices are at 34021 Granada #B, Dana Point, CA 92629. Telephone: (714) 661-9764. Second Class Postage Paid at Dana Point and Santa Ana, CA. **POSTMASTER: Send address changes to I/O NEWS, P.O. Box 17658, Irvine, CA 92713.**

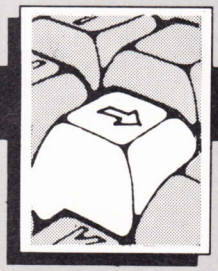
Subscriptions to I/O NEWS are entered with membership in The IACU. Yearly memberships may be purchased for $42 (U.S. delivery address), $49 (delivery address in Canada or Mexico), and $60 (other international delivery addresses). Contact IACU for multi-year membership rates. Back issues of I/O NEWS are available for $10.00 per issue. Please note: all prices are in U.S. dollars.

Return postage must be included with all manuscripts and photos submitted if they are to be returned. The IACU and I/O NEWS accept no responsibility for the return of unsolicited materials.

William E. Jaenicke
Editor and Publisher

Lisa B. Jaenicke
Business Manager

Art Direction: Howard Millman
Printing: Hallmark Litho
Typography: Western Outdoors



INPUT...

Editor:

I own a Cromemco CS2D configured as follows:

- S-100 bus
- DPU (MC68000/Z80A processor)
- 256KZ RAM
- Two Double-Sided Double-Density 5.25" floppy disk drives (390K each)
- PRI parallel interface card (Centronics type)
- CDOS operating system (CP/M version 1.3 and higher compatible)
- VISUAL 500 high resolution (768x585) graphics terminal, emulating Tektronix 4010,4014 terminals & PLOT 10 software in graphics mode and ADM-3A, VT-52, HZ-1500 and DG-200 in conversational mode
- QANTEX 7065 dot matrix graphics printer emulating ANADEX 9620, EPSON MX-80 & MX-100 with Gratrax plus, DEC LA120 & DEC LA34/REGIS graphics and Diablo 630 printers
- Languages: ANSI FORTRAN IV/COBOL, Z-80 ASSEMBLER, CROMEMCO 16K BASIC, CROMEMCO STRUCTURED BASIC, MICROSOFT INTERPRETER/COMPILER BASIC (FORTRAN IV most used for engineering applications)

Taking into account the above configuration, I would be most grateful if you informed me about the following: I know that beyond the supermicros, Cromemco is a leading company in the field of S-100 boards, manufacturing them for several uses. I therefore, wonder if there exists any way to run PC-DOS software on my Cromemco — perhaps by plugging in a card with an 808x processor in the S-100 motherboard. This would be the best for a Cromemco user who wishes to benefit from the wide world of PC software run on such a reliable computer as any of the Cromemco series. Could you inform me about products (software/hardware) providing connection between Cromemco CDOS (or CP/M) and PC software?

Dennis Photopoulos
Civil Engineer
28, Xenias Street
GR-115 28 Athens
GREECE

At this point, I am not aware of any board or software product that will enable Mr. Photopoulos to run PC software under CDOS or CP/M. The hitch is CDOS. There are a growing number of software offerings that enable PCs to be used as terminals, or nodes in a network, but these are tailored for a 68000 Cromix-Plus or UNIX environment. Personally, I wouldn't expect too much in the way of new hardware and software developments for Z-80 based CDOS or CP/M systems. The Cromemco-to-PC connection is, and will probably continue to be, confined to the 68000 based Cromix and UNIX systems. It may be time to consider upgrading your system to Cromix-Plus — where you'll be able to retain your current CDOS software and benefit from performance advantages and the latest in software developments.

It is interesting to note that on the opposite front, there is a good deal of activity directed toward CP/M emulation under MS-DOS. An MS-DOS simulator for CDOS, Cromix, and UNIX, coupled with an appropriate S-100 video controller would do the trick [see next letter] ...

Ed.

Editor:

This letter is in response to the *INPUT* section the *November/December 1985* issue of *I/O NEWS* (Vol. V, No. 1) regarding IBM PC compatibility. I am presently a contract programmer and have the equivalent of a Cromemco System III with a DPU board. For my system to remain practical for me, it must become PC compatible. I have been looking into this problem and believe that I have found some possible answers.

The first level of compatibility between systems is the ability to read and write disk formats from the foreign system. The 64FDC that Cromemco uses is capable of reading a wide variety of disk formats, including that of the IBM PC (this may be verified by attempting a sector read of a foreign disk from RDOS). I am presently working on a program to read and write PC disk files with a similar user interface to Cromemco's CDOSCOPY.

A significantly higher degree of compatibility is suggested by software available for the Commodore AMIGA. The AMIGA is also a 68000 machine and has a simulator program that simulates in software a PC at about 80 percent of the PC's operating speed. While the AMIGA has special purpose hardware that help provide this speed, it should be possible to develop a software simulator for Cromemco systems that will directly execute PC programs. There are also available a couple of PC compatible video cards for IEEE-696 S-100 machines. I believe these are available from Lomas Data Products and Compupro/Viasyn. With these, a complete PC simulator would be possible.

I intend to complete both of the above programs this year. Cromemco S-100 systems are clearly superior to 8088/PC systems; they are both faster and more versatile. I am enclosing an article about a 68020 card to replace the 68000 in existing systems [*see NEW PRODUCTS*];it is reported to allow the above simulator to execute PC programs faster than they will run on a PC! With appropriate support and pricing from vendors, we can expect our S-100 machines to be around for a long time. Anyone interested in the above programs may contact me at the address given below.

Thank you,

Robert S. Coats
2920 Chapel Hill Rd., Apt. 68D
Durham, NC 27707

Thank you!, for both the article and for undertaking the task of programming a PC simulator for Cromemco systems. I have a feeling that such a program would be a BIG hit among Cromemco users, and answer a lot of prayers. Please keep us posted as to your progress.

Ed.

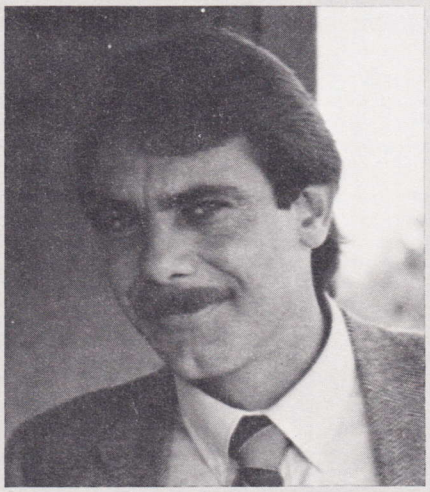
OUTPUT...

Telecommunications. It's a mouthful. And a mindful. A single word encompassing the vast complex of mechanisms and protocols operative in the transfer of digitally encoded information between all types of computers; radiantly delivered over an ever expanding network of cables, telephone lines, and microwave transmissions relayed by satellite. From micro-to-mini-to-mainframe, countless combinations arise as computers link into networks, and as networks network.

You might say that telecommunications is the theme for this issue. Our front cover feature article explores the new Cromemco telecommunications packages, 3270 Bisync and ITI/X.25, which enable 68000 Cromix and UNIX systems to communicate and network with mainframes. The other end of the spectrum — making the link between PCs and Cromemcos — is described from an end-user perspective in the feature articles on ProCall/PC Plus and RDISK.

Elsewhere in this issue you will learn of other exciting developments. For one, we are happy to announce the availability of the I.A.C.U. Software Resource Guide for Cromemco Systems. It is a 180 page software reference source with multiple indexes by product name, software type (broken down into three application groups), and by operating system. Each software package is thoroughly described using a standard format which provides attributes such as the source language, price, product description, number of Cromemco and other installations, how updates are handled, target market, documentation descriptions, training options, source code availability, media supplied on, hardware and operating system requirements, etc.. This first edition was derived from the document that Cromemco has historically made available to dealers only. Now you can have one for your own [*see page 17 for more information*].

Lisa Jaenicke



Bill Jaenicke

Future editions of the guide promise to be even more expansive. To that end, you will find a Software Resource Survey on page 37. I urge you to help by completing the questionaire and returning it to us. We can then proceed to contact the software developers and have them complete a Product Fact Sheet for the guide. If everybody helps, we'll soon have an incredibly diverse database on software applications that run on Cromemco systems — something which can benefit us all.

As expected, we are seeing great strides taking place in the development of software for 68000 systems. In addition to the new telecommunications software, this issue debuts two other packages that promise to have a tremendous impact: the dBIII Compiler for Cromix-Plus and a 68000 Structured BASIC Compiler. On the hardware front, a "piggyback" board to upgrade existing 68000-based systems from a 68000 or 68010 chip to the full 32-bit and speedier 68020 is described in *NEW PRODUCTS*.

For those of you who are hackers at heart, we have a new column — *THE HACKER'S HOME*. It is edited by Rick Dhaenens, the author of last issue's feature article, "CFSU: Cromix File Structure Utility." Rick's insights and expertise in 'C' programming for Cromix and UNIX promise to make this column a goldmine for those who dabble at the systems level.

There have also been some exciting developments closer to home. We recently received shipment of a new CS-100 system, and it was love at first byte (please excuse the pun, I couldn't help myself). I'm still dazzled by the awesome speed and sheer power of it. There's not enough space left here to say more — except that I'm overjoyed! I guess I'll have to make a contribution to the next *USER NOTES* column. But don't let that stop you from contributing something of your own — there's plenty of room for more!

Bill Jaenicke
Editor

ProCall/PC Plus: A User's Perspective

by Ronn Blaylock

There are two kinds of computer people in this world today. First, there are those who open their software package, sit down and thoroughly read the manual first, then install the software and begin to use it. The other type of computer person opens the package, installs the software, tries to use it and then flips frantically through the manual trying to figure out why it won't do what they want it to do.

I fit into the former group. I have a tendency to read everything I can about a software product before I use it. Then occasionally, I find the need to refer to the manual to clear up a point I read about but wasn't entirely sure how to use.

Happily, **ProCall/PC Plus** will work for either type of person. You may operate ProCall/PC using commands, or select items from the menu. The menu is clear and concise. This new version of ProCall is so easy to use, you can run the program with little knowledge of the software within five minutes of putting it on your computer. Or you can spend some time with the manual and learn about all of the powerful features ProCall/PC has to offer.

The hardware I'm using to run ProCall/PC is a MAD-1 with 512K RAM and two floppy drives. I have a Hayes Smartmodem 1200 attached and am also direct-connected to a Cromemco System Three operating under Cromix-Plus. The cabling is switched between the modem and the System Three so I can use either method of communicating with another computer.

The ProCall/PC installation was quite easy. I made a duplicate of the supplied floppy disk, installed it in drive B and typed procall at the system prompt. Then I followed the menu prompts and got my Login: prompt from the Cromix system at 9600 baud. From there I went right to work — just as though I was using a Cromemco 3102 terminal.

The **3102 terminal emulator** is a great feature. I didn't have to worry about having to patch programs such as the Cromemco screen editor to match the IBM monitor. I found I could even run **WriteMaster** without having to make any changes to either system.

The ability to send groups of files between systems (even an entire directory) with one command made short work of moving data between systems. The only problem I encountered was a slight flickering of the screen when using WriteMaster, which I attribute to the color monitor (the flickering didn't exist on the monochrome monitor).

I have a friend who has a tricky system of login to access his computer system. Crosstalk (another communications package) requires that I personally wait for all the prompts and enter all the passwords to get into the system. ProCall/PC came to the rescue with auto-login procedures which do all that work for you. You can put the login procedures in a file or on the command line and let ProCall/PC do the tedious work of logging you on.

Since I communicate regularly with several computers it is a real convenience to have various login batch files, which are called by a simple command such as call greg. ProCall/PC allows me to select the baud rate, send the necessary returns to "wake up" the remote system, log me in through the various passwords, and can even retrieve the files I want, then log me out and hang up the modem. All of these features can be set to function automatically at the command line prompt.

One day a nice lady called to ask if we repaired Persci disk drives. I indicated that we did, and she brought the drive in for repair. While I was chatting with her, she explained that she needed the repair done immediately because she had some end-of-the-year reports which had to go out. A quick look at the drive told me a hard to get IC had failed and it would take a week or more to get that particular IC.

It seems that this young lady was running CP/M double-sided double-density disk format and using Wordstar to do her work. She also had IBM PCs at work. I told her that I could copy her diskettes to MS-DOS format and with minor changes to her Wordstar files she could print her reports from an IBM PC.

Normally, it's a long process to go from CP/M 8 inch DS/DD to MS-DOS 5 ¼ DS/DD diskettes. I first had to format several diskettes for single-sided single-density CDOS. Then I had to boot CP/M on a Cromemco System Three and copy the DS/DD CP/M to SS/SD CDOS.

ProCall/PC came to the rescue from here. Jeff McNaught, of PSD, was kind enough to provide me with a preliminary release copy (this version is now available) of ProCall/PC Plus which included **Kermit**, **Rfile**, and **Sfile** capabilities allowing me to do multiple file transfers. The rest of the job of transferring over 200 files was easy.

I booted the MAD system, loaded ProCall/PC, and logged into the Cromix system. I then created a temporary directory, did a cdoscopy from the CDOS diskette into the temporary directory, and finally selected the "transfer file" option in ProCall/PC with the "Rfile/Sfile" option.

After selecting that one command, in under 10 minutes I had duplicates of all the files in IBM format on the MAD system. The process was quick because I was able to transfer the files via direct connect at a full 9600 baud. There was approximately 230K of data in the directory.

As you can see, ProCall fills a significant need in the computer communications area. Now that ProCall/PC Plus has ambiguous file transfer, when used with a Cromix system, it can be an extremely powerful tool to attach PC's as workstations to form a diverse multi-user network. It gives a person the best of both worlds without having to go out and buy a new computer system or having to buy an expensive network to share already existing resources.

As an add-on, I might say that I have used ProCall/PC on a 3Com Server to a Cromix system and used the letter quality printer on the Cromix system instead of having to buy a new printer for the network.

The new release of ProCall/PC has the Kermit file transfer protocol. It adds a whole new world of communications for both Cromix and IBM machines. We will now have the ability to talk to almost any machine in the world from our offices, assuming, of course, that you can connect to a modem at both ends.

S-100 ProCall, and now ProCall/PC, is, and has been my choice for communication between various computer systems ever since ProCall was first released. Over the few years of its existence, enhancements have been added which have increased its effectiveness for a multitude of situations where good, effective, and easy-to-use communications were needed between formerly incompatible machines. Without it, I don't know what I would have done!

About the Author:

Ronn Blaylock *is currently Service Manager for* MCM Enterprises *in Palo Alto, CA. He has 5 years experience in software and 4 years experience in computer hardware and repairs with an extensive background in field troubleshooting throughout the San Francisco bay area, and has a strong background in CDOS, Cromix, Cromix-Plus, UNIX, PC-DOS, and MS-DOS.*

Before entering the computer field he spent 20 years in the Navy gaining management skills as supervisor for various departments throughout his career.

RDISK:
Virtual DOS Disks Under UNIX

by David Saul

Today the IBM personal computer (PC) has achieved worldwide acceptance as a standard for personal and small business computing. With over 4 million IBMs and another 1 million compatibles installed, these machines are found in every aspect of computing from terminals on large mainframes to single station word processors. The DOS operating system is second only to APPLE-DOS in the number of software applications available. The current decline in the price of the PCs (a PC soon will cost less than a smart terminal!) will firmly establish it as the industry standard.

The advantages of having a standard machine with a standard operating system are:

1. Software houses only have to support one version of their product.
2. Floppy disks from one machine can be read on another.
3. A large variety of standard hardware options are available.

The main disadvantage of having a standard is that innovative technology not consistent with the standard is discouraged. We therefore have to live with machines that are not state of the art until a new standard is adopted.

The two main disadvantages of the IBM PC are: it is a single user machine, and it has a maximum disk size of 32Mb.

The current drive of the PC industry is to "work around" these deficiencies. These solutions usually have some unwelcome side effect.

To work around the single-user limitation, networking has appeared. A network typically connects a number of PCs on a bus, ring, or star hierarchy — assigning each PC (or node) an address.

To communicate from one node to another you test to see if the network is busy and then send out your message with the appropriate address. This message could be "Go to PC 47 file D:\DATA\MASTER.DAT record 1634 and write "2334,JONES,3,337.55." The problem is that the file MASTER.DAT could be in use by ten other PCs and the network and/or the software may not offer any multi-user protection. The main difference between a network and a true multi-user system is no file or record locking in the operating system. To fix this problem the software package must be modified for each type of network (there goes our standardization!) or the file must be locked so that only one user can access it at a time (there goes our multi-user capability!).

To get around the maximum disk size we can change the DOS block size. For example DOS uses 512-byte blocks with a limitation of 2^{16} blocks for a disk size of 32Mb. If you make the block size 8192 bytes, then the largest partition becomes 512Mb. The problem is that files are created in an even number of blocks. A 1 byte file would use 8192 bytes of disk space!.

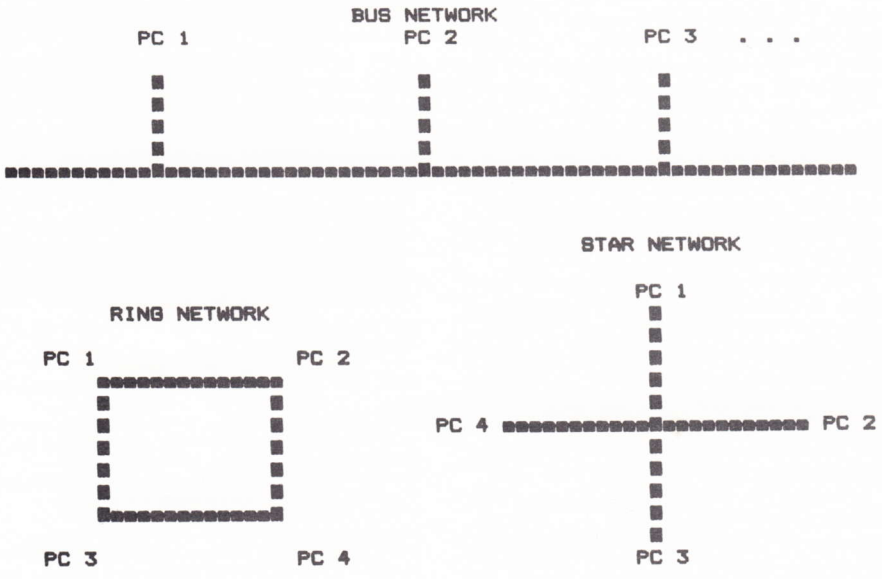
One possible solution is to connect the PC running DOS to a multi-user system running UNIX. Several software packages are available to connect a PC to UNIX as a terminal. These packages allow up and downloading of files and piping input and output, in addition to all the standard UNIX features. Basically, they are terminal emulators. To expand the capabilities of our PC we need a virtual disk server.

RDISK from **MODULAR System Design** allows a PC connected to UNIX to use any disk mounted to UNIX as a virtual disk under DOS. If you have 1200Mb under UNIX you can run a DOS program using 1200Mb. All you need is a PC with 128Kb of memory, a single floppy, a serial communications port and of course a UNIX system. This PC today lists for under $1000. What RDISK does is to format a section of the UNIX disk to be in DOS format. The PC is connected to UNIX as a terminal and sees UNIX as up to 16 virtual disks letters E: through T:. Drives A: to D: are reserved for the DOS physical units. The virtual disks to UNIX are simply large files and can be backed up by cp or tar.

RDISK can run as fast as the UNIX system can accept data. Installations running at speeds over 100 kilobaud are in use. To illustrate RDISK the following sample commented session is provided. ►

The examples shows how UNIX can be used as a **virtual file server** for a PC-DOS computer. Note that all of the UNIX security features are still in place. We have the login with password and the owner, group, and world access list for the individual files. The newest release of UNIX supports file and record locking which protects multiple RDISK users who are accessing the same file. The most important feature of RDISK is that it is completely transparent to the PC user. No knowledge of UNIX is required. From the PC user's point of view, he sees only 16 very large disks attached to his system. All standard single user DOS software can now be run by multiple users sharing a single disk.

So, with the marriage of DOS and



```
MS-DOS 3.1                | Booting DOS
RDISK Version 1.0

A>TERM 9600 1             | Connect to UNIX at 9600 baud
                          | on PC port 1
UNIX Version 5.2          |

Login:guest               | Log in to UNIX
Password:

Welcome to Cromemco System 400

UNIX 5.2
%speed 57600              | Change baud rate to 57.6kb
%rdisk                    | Bring up server

RDISK Version 1.0 Running

A>                        | Back to DOS

A>host fmt -s 32767 -v PITTSBURGH /usr/guest/bigfile
                          | Format a file called bigfile on UNIX with
                          | size 32 MB and VOLID PITTSBURGH
A>host fmt floppy         | Format a file called floppy to default
                          | size 360kb

A>mnt E /usr/guest/bigfile | Mount drive E to file bigfile

Drive E mounted

A>mnt T floppy R          | Mount drive T to file floppy read only

Drive T mounted

A>sts                     | Check virtual disk status

Available drives are E to T

Drive       STATUS        FILE
-----       ------        ----
  E         READ/WRITE    /usr/guest/bigfile
  T         READ ONLY     floppy

A>COPY A:123.* E:    | COPY LOTUS 123 TO VIRTUAL DRIVE E
123.EXE
123.HLP
123.CNF

  3 FILE(S) COPIED

A>E:                      | Move over to UNIX on virtual drive E
E>DIR                     | Do a directory of virtual drive E

 Volume in drive E has label PITTSBURGH
 Directory of  E:

123      EXE    89856   1-01-80  12:04a
123      HLP   113416   6-07-83   1:23a
123      CNF      256   1-01-80  12:01a

        3 File(s)  33528832 bytes free   | Note space available!

E>A:                      | Back to physical drive A
A>HOST who                | Run the UNIX who command from DOS

root      console   Jan 4 10:40
guest     tty1      Jan 4 12:10
david     tty8      Jan 3 10:03
sean      tty9      Jan 4  9:02

A>COPY CON T:DOCUMENT     | Copy the console to file DOCUMENT
HI THIS IS A TEST FILE!
^Z

Write protect error writing Drive T | Remember T was mounted
                                    | READ ONLY!
Abort, Retry, Cancel ?A
```

Continued

UNIX a PC user can now:

1. Run UNIX applications using the PC as a terminal,
2. Run DOS applications using the UNIX hard disk,
3. Copy files from DOS to UNIX and back,
4. Print from DOS using the UNIX printer and from UNIX using the DOS printer.

It may not be a marriage made in heaven — but made in Pittsburgh is close enough.

About the Author:

David Saul *is President of* MODULAR System Design, *a firm founded in 1977 which specializes in the development of custom communications for large corporations. Mr. Saul was one of the lecturers at a recent communications class sponsored by Cromemco, and held in Frankfurt, Germany. In attendance were Cromemco dealers and OEMs representing fourteen countries.*

For further information about RDISK contact:

MODULAR System Design
6425 Darlington Road
Pittsburgh, PA 15217
(412) 521-6700

RDISK *Continued*

```
A>copy con E:DOCUMENT
This is a TEST Document!
^z

1 file(s) copied

A>host                                  ! Go to UNIX C-Shell

%ls -l                                  ! Do a detailed list

-rwxrwxrwx  1  guest  guest   33554432 Jan 4 10:4 PITTSBURGH
-rwxrwxrwx  1  guest  guest     368640 Jan 4 10:3 floppy

%dir PITTSBURGH                         ! A DOS dir under UNIX

 Volume in drive E has label PITTSBURGH
 Directory of  E:

123        EXE     89856   1-01-80  12:04a
123        HLP    113416   6-07-83   1:23a
123        CNF       256   1-01-80  12:01a
DOCUMENT               24   1-04-86  10:10a

        4 File(s)  33528832 bytes free

%extract PITTSBURGH DOCUMENT  ! Convert a DOS file to UNIX format
This is a Test Document!      ! This output could be piped
%exit                         ! Exit the C-Shell

A>setprt lp                 ! Set DOS printer lpt3 to UNIX printer lp
A>copy DOCUMENT lpt3        ! Print DOCUMENT on UNIX printer
A>rcopy DOCUMENT /usr/guest/doc ! Copy DOCUMENT to UNIX file doc
A>umnt T                    ! Unmount Drive T

Drive T has been dismounted

A>umnt E                    ! Unmount Drive E

Drive E has been dismounted

A>RXIT                      ! Halt server and log off UNIX

RDISK processor halted
```

Can A Cromemc

by Pat Maguire

I recently visited a company that had just purchased an eight year old minicomputer, and therein lies a tale.

This company had an older mini for several years, with mail list and invoicing software. The company buys and sells large mail lists. They purchased a Cromemco Z2-H about three years ago. In late 1983, they called ACS looking for accounting software. Our Accounts Payable, Receivable, and General Ledger packages were installed. Several changes were made due to the deferred income nature of their business. A few other changes were made over the next year and a half, but essentially a steady state existed.

Another company in the same business was closed, leaving a minicomputer available for less that ten cents on the new price dollar. They bought the system for $5000, with software. This appeared to be a super value on the surface. Unfortunately, the maintenance for this rather unreliable, old machine, which was to replace their older mini, was about $10,000 per year. I pointed out that we (ACS) could provide a new, reliable, expandable Cromemco on a lease, with maintenance, for less than maintenance alone on their bargain machine. This was 2 megabytes versus 64K, 140 meg disk versus 128 meg, and a new 9 track tape versus the older one (same manufacturer).

We compared speed on the mini, which uses the Pick operating system. A search of 30K records of 140 bytes each was run on both machines. It took 5:50 on the Cromemco, using Informix, and 7:10 on the big machine. In addition, the "wall heater" (our ungracious name for that big machine) required a full room with a special spot air conditioner. The Cromemco could fit on a desk. They get a room for free! One last point — if I may beat a dead horse — the building is subjected to power outages from time to time. A UPS (Uninterruptible Power Supply) for the big machine costs about ten times more than one for the Cromemco.

The sad thing is, they could have done this in the beginning and saved more. Now, they may not do it at all.

I think a major point exists here. Many potential and existing customers do not think of Cromemco when they thing of serious business applications. I suggest a series of articles in I/O NEWS by users, dealers, etc., pointing out that a Cromemco can do it, in those areas normally considered mini territory. I will describe one of our largest installations as a case in point.

Auburn Brass, a division of Amerock, an Anchor Hocking Company, is a producer of very high quali-

co Do It?

ty bathroom fixtures and accessories. They procure components from around the world for processing in two plants in Southern California. This processing includes plating, buffing, assembly, testing, and inspection.

Their initial step into computers was an inventory package on a Z2-H under CDOS. The inventory was not satisfactory, but the original vendor would not make revisions and did not provide systems documentation (e.g., file formats).

A payroll system was purchased from ACS in December, 1982. Accounts Payable, then Accounts Receivable, followed. One change to A/R was the creation of aged invoices by territory. Payroll was enhanced to allow 401 (k) deductions, and additional personnel reports. A few other changes were made to these packages. The ACS Mail List Management System was added. A special program created the initial data file by scanning the A/R Customer file.

A major concern of the company was cost control and reporting. Various finishes, such as gold plating, affect the costs sharply. A custom cost system and multi-level bill of materials (BOM) was developed. A merchandise transfer document is used to move parts between locations (purchased parts, WIP, etc.). The cost system evaluates the material listed on these documents. Various reports provide audit trail and documentation for personnel from the company, as well as the parent company and outside auditors. Since invoices were being entered to report costs of goods sold, this sub-system was used to create commission reports.

The hardware had by now been expanded to three terminals and two printers, operating under Cromix.

With a BOM, it was now possible to develop an explosion of production lot assemblies into the detail required for the lot. We called this a production requirements report.

At this point, the previous inventory system was replaced with one designed for the company's special requirements, including costs by various categories. Receiving reports, merchandise transfers, invoices, and returned goods authorizations all interface with the costing and inventory systems. Reports provide receipts and issues information. Additions to inventory master records (e.g., new part numbers) are performed automatically from the documents. The computer compares all information to the BOM for validity at input.

The company had now grown substantially. A Cromemco 68000 System III with a 50 meg SMD drive was added, along with three more terminals. This occurred shortly before the 50 meg with STDC was introduced.

A special locator system was implemented. This tracked items in the finished goods warehouse by location.

Order entry was then implemented. This allowed us to tie everything together. The order entry system accumulates sales data, creates requirements reports, prints sales orders, back orders, and invoices, and updates A/R and costing. Pricing is an important aspect of order entry. Retail prices, which vary with finish, are stored in the BOM files, along with a product code. A discount code is stored in a customer file. These two codes are used to scan a user defined matrix to obtain the customer's discount percentage. Finishes and colors are compared to user definable tables during input to assure validity. Stock descriptions, customer bill-to and ship-to name and address data, and all other feasible fields have default entries.

Some numbers from the installation are informative. A/R maintains approximately 1,200 customers and 3,500 open invoices. A/P maintains 300 vendors and receives about 300 invoices per month. Over 100 W-2 forms exist in payroll. About 200 orders are processed per week, comprising some 800 line items. The bill of materials contains over 4,000 records, and inventory about 7,000. Close to 14,000 transactions (merchandise transfers, etc.) were posted to inventory last month. The last physical inventory totaled 15,000 records.

Because of the amount of customization required, a set of utilities is used. This allows faster revisions and creation of sub-systems. About 90% of the input and edit functions are supported by utilities driven by parameter files. Print routines are highly standardized. All major files are keyed, and the keys are kept in RAM disk. Command files move the key files from and to the hard disk during startup and shutdown. Rebuild routines allow re-creation of the keys in the event of non-standard shutdown. RAM disk greatly increases the processing speed, especially BOM explosions. A small sample explosion required four hours manually, but is handled in less than ten minutes on the computer. One unique aspect of an explosion in this installation is the need to track finishes and colors. Flags in the BOM files tell the system whether an item has a finish or color. Seals, for example, have neither. As a side note, two people tried their hand at the sample explosion, and each made different mistakes.

Of course, accuracy has increased over the manual system, and the appearance of orders and invoices has been improved. The order handling paperwork cycle has been reduced by at least two days, while various reports have helped reduce the material cycle.

While ACS has a much simpler BOM/inventory system available, I feel that this complex one better illustrates the point. This is the kind of system that would have been considered out of the question on a micro a few years ago. I look forward to seeing other examples from other readers which illustrate that "a Cromemco can do it."

About the Author:

Pat Maguire *is President of* Advanced Computer Software, Inc., *a firm specializing in standard and custom software packages for Cromemco systems, founded in 1980.*

With a Masters in System Engineering, Maguire has been programming since 1959 — with programming experience spanning mainframes to micros. He has been working with Cromemco microcomputers since 1978.

Advanced Computer Software, Inc.
700 S. Tustin Ave., Suite B
Orange, CA 92667
(714) 771-0852

Telecommunications

Continued from front cover

chronous form (such is used to connect your CRT terminal to your computer) in that a start bit and stop bit are not required for every character. This means that, for a given bandwidth telephone line or other physical link, data can be communicated at a faster rate synchronously than it can asynchronously. Both the Cromemco **Biart** and **Quadart** can support synchronous communication. The RS-232 synchronous output of either the Biart or Quadart can be connected to a synchronous modem for transmission over a telephone line.

Bisync

The first widely adopted synchronous communication standard was called **bisynchronous communication** and is also known as BSC or Bisync. Bisync was developed by IBM in 1967, and was first used to connect so-called remote batch terminals (such as card readers) to a mainframe computer.

The first IBM remote batch terminal to use the Bisync data link was the model 2770. This terminal was designed to read and transmit punch card files to a remote computer, and to receive from the remote computer files that would be punched onto cards or printed on a line printer. The model 2770 was succeeded by the model 2780 and model 3780. Other such IBM products were the 2968 (a magnetic tape unit) and the 3741 (a key-to-desk machine).

Cromemco computers have, for some time, been able to emulate such remote batch terminals by using the **IOP Quadart** hardware with the **RBTE** (Remote Batch Terminal Emulator) software. The RBTE software can operate under the CDOS or Cromix operating system at data rates of from 1200 to 19,200 baud.

Later IBM introduced a class of remote **interactive** (as opposed to batch terminals. These were known as the model 3270 terminals. Cromemco computers can now emulate 3270 terminals by using the **Biart hardware** with the new **3270-BSC software**. Versions of 3270-BSC are available for both the Cromix and UNIX operating systems and support transmission rates of from 1200 to 9600 baud. In particular, a Cromemco system running 3270-BSC

Table 1

CROMEMCO SERIAL INTERFACES

MODEL	NUMBER OF SERIAL CHANNELS	INTERFACE STANDARD	SERIAL PROTOCOL	SPEED	MAJOR USE
OCTART	8	RS-232C	ASYNCHRONOUS	50 — 38,400 BAUD	Connecting a Cromemco computer to CRT terminals, serial printers, or asynchronous modems
BIART	2	RS-232C RS-422 RS-423	ASYNCHRONOUS	30.5 — 19,200 BAUD	Connecting a Cromemco computer to synchronous modems
			SYNCHRONOUS	0 — 1,000,000 BAUD	
			e.g.	BISYNC (used in IBM 3270 protocol) SDLC (used in IBM SNA protocol) HDLC (used in CCITT X.25 protocol)	

GLOSSARY

RS-232C	The standard adopted in 1969 to describe the connection between a computer or a terminal (called the DTE) and a modem (called the DCE). Limited to 19,200 baud and 50 feet of cable.
RS-422 and RS-423	Standards adopted in 1978 to replace RS-232C when higher speed transmission (up to 10,000,000 baud with 40' cable) or longer cable lengths (up to 4000' at lower baud rates) are required.
ASYNCHRONOUS MODEM	A modem used for low speed data transmission — typically up to 1200 baud on a dial-up telephone line.
SYNCHRONOUS MODEM	A modem used for high-speed data transmission — typically 2400 baud or faster on a dedicated line.

Table 2

SYNCHRONOUS COMMUNICATIONS PROTOCOLS

SOFTWARE	INTERFACE HARDWARE	OPERATING SYSTEM	IBM DEVICES EMULATED	LINK-LEVEL PROTOCOL	BAUD RATE SUPPORTED
RBTE	IOP QUADART	CDOS or CROMIX	2770, 2780, 3780, 2986, 3741	BSC	1,200 — 19,200
3270-BSC	BIART	CROMIX or UNIX	3278, 3287, 3274	BSC	1,200 — 9,600
X.25	BIART	CROMIX or UNIX	—	HDLC	1,200 — 9,600

GLOSSARY

RBTE	— Remote Batch Terminal Emulator
BSC	— Bisynchronous Communication
SDLC	— Synchronous Data Link Control
HDLC	— High-level Data Link Control

software and connected to a mainframe over a synchronous modem will have its terminals *appear* to the mainframe as IBM 3278 Model 2 terminals and its printers appear as IBM 3287 printers.

SDLC/HDLC

As time went on, it became clear that there could be improvements to the BSC data link protocol. In **1974** IBM announced such an improvement called **Synchronous Data Link Control**, or **SDLC**. SDLC and its derivatives are the modern communications standards for the high speed communication of computer data. When the International Standards Organization (ISO) adopted a data link protocol it based it on SDLC and called it **HDLC** (High-level Data Link Control)

The Cromemco Biart serial interface, in addition to supporting BSC, also supports SDLC and HDLC. (See accompanying Table 1 for Biart and Octart specifications).

X.25

When only two devices are linked together, the data link software (such as BSC, SDLC, or HDLC) is all that is required to coordinate the transfer of information. If more than two devices are connected in a network, however, a higher level program is required to coordinate the flow of the packets of data so that when data is sent from one device it is properly routed to the intended receiving device. One such type of networking software is called **X.25**. The X.25 standard was published by the International Standards Organization in 1980. The link-level protocol for X.25 is HDLC.

Cromemco computers can now connect to X.25 networks with the **Biart** hardware and the X.25 software (model ITI/X.25). Terminals on the Cromemco computer become **interactive terminals** on the X.25 network. For this reason the software is said to provide an **interactive terminal interface** or **ITI**. Since the function of the software is to assemble and disassemble packets of information that are on the network, it is said to provide **Packet Assembler Diassembler** or **PAD** capability.

The X.25 capability on Cromemco computers is good for one of two purposes. First, it can provide access to **public** X.25 networks. **Telenet** and **Tymnet** are the two best examples of these. Second, it can allow a customer to construct a private X.25 network. In Korea the Air Force has 60 such Cromemco systems in a network to communicate strategic information between air bases.

Cromemco systems support X.25 data transmission rates of from 1200 to 9600 baud.

Summary

In addition to RBTE Cromemco now supports two important new telecommunications standards. A summary of these capabilities is shown in Table 2. With this new software Cromemco computers, under either Cromix or UNIX, can communicate with mainframes and become fully integrated into an information system.

Continued on page 20

dBIII Compiler

Continued from front cover

piled applications on the 68000 for increased speed.

dBIII Compiler features not available to the Cromemco dBASE II user include:

1. Up to ten files open simultaneously in one of ten separate work areas versus two in dBASE II.
2. Date field support.
3. Sort on multiple fields.
4. Public and private memory variables and parameters.
5. Running operating system commands from within dBASE programs.

dBIII Compiler for Cromix-Plus supports a well featured interface to the Cromix-Plus operating system including:

1. Record and file locking for multi-user protection. The record protection is dBASE record oriented, not "sector" oriented like other protection schemes. This means you lock only the data you need to lock and other users are only momentarily delayed while you have that one record selected. The file protection ensures you don't update a dBASE file header while someone else is using the file. It also ensures that indexing is not being performed by two users simultaneously.
2. Full pathnames for files.
3. Spooling of printed output.
4. Reading of the termcaps file for terminal capabilities to allow the use of many different terminals without using install procedures.

dBIII Compiler for Cromix-Plus has maintained data file compatibility with dBASE II and dBASE III. dBIII Compiler compensates for the fact that the normal byte order for words in these files is reversed from that desired by the 68000. This means that you can use all of your present data files and convert to dBASE III format when you wish. It also means you can continue to manipulate data with dBASE II (though we don't know why you would) and yet use compiled applications on the same data. In addition, only minor changes are required in your present .CMD files to convert them to .PRG files for dBIII Compiler.

Finally, dBIII Compiler for Cromix-Plus gives you all the above features with the speed of a compiled program running on the 68000. We believe you will find your applications running much faster, especially in situations involving multiple changes to selected records. Indexing runs at approximately the same speed as dBASE II under Cromix-Plus (Cromix-Plus already does such a good job of file buffering).

Overall, we believe you will be excited over the productivity gains dBIII Compiler for Cromix-Plus can offer you.

About the Author:

Rich Mitchell *is a specialist in the art of 'C' programming currently working with* Software Standards, Inc., *a two year old firm which ports software for Cromemco systems. Mitchell did the port of WordTech Systems' dBIII Compiler for Cromix-Plus.*

Software Standards, Inc.
6191 Choctaw
Baton Rouge, LA 70805
(504) 355-8024

South America

Continued from front cover

ple of the contribution microcomputer technology is making to the economic growth of developing nations. Microcomputer systems can provide enormous leverage of existing resources in these countries, just as mainframe computers do in much larger economies. In South America the dependability, expandability, and power of Cromemco supermicrocomputers have positioned Cromemco as the leading surrogate mainframe in a dazzling variety of applications.

To appreciate the economic contribution of Cromemco computers in these countries take, for example, the textile mills of Medellin, Colombia. Although once competitive in world markets, the aging textile production machinery now cannot produce fabric with the same quality in the same volume as more modern, automated equipment. Juan Henao and Alvaro Perez decided to change that. Certainly re-equipping the vast textile factories with more modern equipment would not be economically feasible. But why not outfit the existing machinery with the requisite sensors, actuators, and a microcomputer control system to bring the machines up to modern standard?

Both Juan and Alvaro were educated in Colombia, and then attended graduate school in the U.S. Juan attended the University of Michigan, and Alvaro attended Stanford University where he was first introduced to Cromemco equipment. Now operating a Cromemco distributorship, **Control Sistematizado**, in Medellin, they won a major contract from Pantex, the country's leading textile producer, to automate their factory. Jaime Blandon of Control Sistematizado was given responsibility for project implementation.

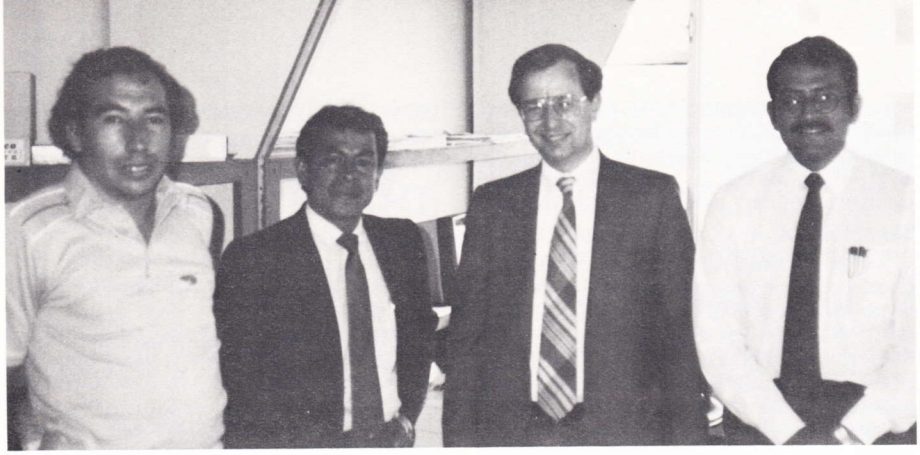
Their approach was this. Cromemco

single card computers (SCC), memory, 8PIO cards, and ACD12 interfaces were mounted in CC-8 card cages. These card cages were powered by Cromemco PS-8 power supplies. Each of these SCC-based modules was used to control a specific process in the manufacturing equipment, e.g. fluid levels, temperatures, humidity, fabric tension, and run rate. A central Cromix system was then used to monitor each of these individual processes and to drive the master control console. As a result of this innovative system, Pantex is already enjoying an increase in the consistency and quality of their product and is anticipating actually increasing their manufacturing rate by as much as 20% in the near future. This translates into an enormous productivity increase, and provides a new and exciting competitiveness for Colombia in the world textile market. And this is just one example of how Cromemco computers are increasing productivity in South America.

Chile, for example, relies increasingly on the export of agricultural products in its foreign trade. Computer-aided research in the genetic engineering of crops, agricultural land use, and other areas of agronomics is critical to maintain Chile's agricultural leadership in world markets. The institute responsible for this important activity is the Instituto de Investigaciones Agropecuarias located in Santiago.

This institute analyzes data from its experimental stations, carries out agronomic analysis, and in these and other related activities relies heavily on the latest in computer technology. Working closely with **EPROM**, Cromemco's distributor in Chile, the institute has installed over a dozen Cromemco computers. According to Mr. Fernando Silva of the institute, since their work includes both scientific research and economic analysis it was important to standardize on computer systems that could run both scientific languages (primarily Fortran and Basic but with some work in Pascal and C) as well as business languages (primarily COBOL). It was also important to have high processing speed for modeling work, and to have the ability to store large databases. Cromemco computers fit the bill. The importance of the role Cromemco computers have played in this critical national research is attested to in the institute's conference room. Hanging on the wall is not, as you might expect, a picture of wheat fields or grape vineyards, but rather, a picture of a Cromemco computer.

Interestingly, Dr. Jorge Bellet, founder and managing director of EPROM, has as a strategy specifically focussed on applications of Cromemco computers in Chile that can leverage the considerable national resources of his country. For this reason several key members of EPROM's technical staff come not from a computer background, but rather from a strong scientific applications background. Dr. Carlos Infante, who holds a Ph.D. in physics



Dr. Harry Garland with Compusystems staff in Quito, Ecuador. From left to right: Jaime Prado, Marcello Izurieta, Harry Garland, and Fernando Rivera.

from the University of Oxford, heads EPROM's Technology Applications Division, and Carlos Thuene, educated in Germany as a geologist, heads the Industrial Minerals Division. Dr. Bellet's wife, Monica, is also a physicist and is active in the business and marketing activities of EPROM.

In keeping with this strategic thrust EPROM's customers include, in addition to the prestigious Instituto de Investigaciones Agropecuarias, such important institutions as the Instituto Interamericano de Cooperacion para la Agricultura, the University of Chile, the Catholic University, and the Compania de Acero del Pacifico.

EPROM has also developed extensive commercial application software, under Cromix, that is specifically tailored for the Chilean market. For 1986 Dr. Bellet reports that his major emphasis will be on the System 400, which he believes can take a very large share of the present minicomputer market in Chile.

The variety of uses for Cromemco computers in South America is as varied as the land itself, showing the great versatility of these machines. Imagine, for a moment, the enormous data base required to store all the listings for the telephone books of the entire country of Ecuador. Surely the task for a mighty mainframe computer, wouldn't you think? Wrong. The listings are actually stored on a Cromemco System 400 running under UNIX.

The software for this telephone book applications was written by Roberto Gaio, general manager of **Infotron** in Quito. Infotron worked under contract to **Offsetec**, the major publishing house in Quito with responsibility for printing the telephone books for Ecuador's two major cities: Quito and Guayaquil. At Offsetec, the phone book program is under the general management of Rodrigo Grijalva, who purchased the first System 400 in Ecuador.

Offsetec's use of Cromemco for the phone book project is just one of a series of successful applications they have had for Cromemco computers. Another major part of their business is the printing of bank checks. This is carried out on a special Dataproducts printer capable of printing 30,000 checks per hour. The printer is controlled by a Cromemco System Three. This check printing system, including the Cromemco computer, is marketed by Dataproducts throughout South America. In addition to installations in Ecuador, which are distributed by Optronic in Quito, check printing systems using Cromemco computers are now also in use in Peru and Argentina.

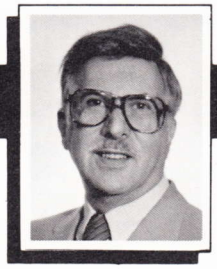
Marcello Izurieta, general manager of **Compusystems** in Ecuador, emphasizes commercial applications for Cromemco products in his country. In addition to major installations at companies like Ruben Herrmann and Offsetec, Compusystems sells EDP services for smaller businesses. The EDP services division uses a Cromemco System Three, with business application software developed specifically for the Ecuadorian market by Compusystems. Mr. Izurieta points out that as his EDP services customers grow and require an in-house computer system, they can then very easily transition to an in-house Cromemco system with no software conversion required. As a result of this business focus Mr. Izurieta and his colleagues at Compusystems have made Cromemco one of the leading names in business computers in Ecuador. In fact when you check into the elegant Hotel Colon International in Quito, don't be surprised to see a Cromemco C-10 in use behind the registration desk.

Some applications for Cromemco computers in South America are similar to applications in the United States, while others are not. Watch television in Colombia or Brazil, for example, and you will see computer-generated weather maps and other graphics generated by a Cromemco graphics system, just as in over 400 television stations in the U.S. The strong market for Cromemco computers in the U.S. government can also be seen reflected in the government markets of other countries. Three System 400's are used by the Department of Commerce in Colombia and connect to an IBM mainframe using Cromemco's 3270 software. Other applications, such as the monitoring system for a huge coffee processing plant being developed by Control Sistematizado or the telecommunications system being installed by TCC in Peru, are unique to the local economies. In all these applications, though, Cromemco computers are making a real contribution to improving productivity, competitiveness, and the standards of living of these countries.

Current Versions of Cromemco Software

This table lists the current versions of recent Cromemco software. It was derived from Cromemco's Software Product Version Report of February 15, 1986. The following notations are used: "NA" implies that the information is not applicable or was not supplied in the product version report. Models ending with -'**D**' are for 68000 Cromix; those ending in '**X**' are for 68000 UNIX. Almost all software is supplied on both 8 inch and 5 inch diskettes, so the "L" (for large) and "S" (for small) have been omitted from the model numbers. Software packages which have been updated since the last printing of this list (Vol. IV, No. 6) are printed in **boldface**.

MODEL	PACKAGE	RELEASE	VERSION	CREATED
3270BSC-D	**3270 Bisync Communications Software**	**1**	**03.05**	**02/21/85**
3270BSC-X	**3270 Bisync Communications Software**	**1**	**NA**	**07/25/85**
ANI-D	68000 COBOL Animator (Debugger)	2	NA	02/10/86
ANI-X	68000 COBOL Animator (Debugger)	1	NA	04/25/85
ASM-D	68000 Macro Assembler	2	01.14	02/16/83
BAS-D	**68000 BASIC Interpreter**	**2**	**02.40**	**08/20/85**
BASLIN-D	**Baseline Graphic Software (for S-series)**	**1**	**02.01**	**06/28/85**
BAS-X	68000 BASIC Interpreter	1	02.20	09/27/84
BNET-X	**B-Net Ethernet Communications**	**2**	**NA**	**01/27/86**
C10CPM	C-10 CP/M Operating System	1	02.00	01/17/84
CAMR	CalcMaster	4	NA	02/29/84
CCC	Z-80 'C' Compiler (Cromix only)	2	05.10	01/04/83
CCC-D	**68000 'C' Compiler**	**4**	**02.41**	**08/16/85**
CCC-X	68000 'C' Compiler	1	02.20	09/17/84
CISAM-D	**C-ISAM File Access**	**2**	**2.04**	**08/21/85**
CISAM-X	**C-ISAM File Access**	**2**	**2.04**	**07/16/85**
COB-D	**68000 COBOL Compiler**	**3**	**NA**	**04/30/85**
COBRT-D	68000 Run-Time COBOL	1	NA	05/02/85
COBRT-X	68000 Run-Time COBOL	1	NA	05/21/85
COB-X	68000 COBOL Compiler	2	NA	11/04/83
COLL	Cromemco Overlay Linker	3	02.04	03/25/83
CONFIG-X	**UNIX Configuration Software**	**2**	**NA**	**01/27/86**
CRO-D	68000 Cromix Operating System	8	20.65	03/27/85
CROMIX	Z-80 Cromix Operating System	11	11.27	07/03/84
CRO-PLUS	**68000 Cromix-Plus Operating System**	**5**	**31.05**	**11/12/85**
C10-CPM	CP/M Operating System for C-10	1	02.20	01/17/84
CXDR	**68000/Z-80 Cromix I/O Drivers**	**2**	**NA**	**01/16/86**
DIMR	DiskMaster	2	01.11	09/08/84
DOS	CDOS Operating System	12	02.58	11/07/83
ESQL-X	**Informix ESQL**	**1**	**NA**	**NA**
FDA	Z-80 Macro Assembler	12	03.10	07/18/83
FDC	Z-80 COBOL Compiler	6	04.64	03/29/83
FDF	Z-80 FORTRAN IV Compiler	11	03.42	03/30/83
FDR	Z-80 FORTRAN IV with RATFOR	4	01.05	03/29/83
FM2-D	68000 COBOL Forms-2	1	NA	10/24/83
FM2-X	68000 COBOL Forms-2	1	NA	04/25/85
FOMR	FontMaster	5	01.16	08/19/83
FOR-D	**68000 FORTRAN 77 Compiler**	**7**	**02.41**	**08/16/85**
FOR-X	68000 FORTRAN 77 Compiler	1	02.20	09/17/84
FSTBAS-D	**68000 Fast BASIC Compiler**	**2**	**02.40**	**08/20/85**
FSTCCC-D	**68000 Fast 'C' Compiler**	**3**	**02.41**	**08/20/85**
FSTFOR-D	**68000 Fast FORTRAN Compiler**	**4**	**02.41**	**08/20/85**
FSTPAS-D	**68000 Fast PASCAL Compiler**	**4**	**02.41**	**08/20/85**
INFX-D	**Informix Relational Database**	**3**	**03.30**	**08/21/85**
INFX-X	**Informix Relational Database**	**2**	**03.30**	**06/18/85**
KSAM	KSAM File Access (Z-80 Cromix)	3	01.04	03/01/83
LSP	LISP Interpreter	5	01.08	03/31/83
MacLine	**Apple MacIntosh to UNIX Communications**	**1**	**NA**	**NA**
MAXASM-D	Maximizer Microcode Assembler	1	02.08	11/06/84
NCG-X	68000 COBOL Native Code Generator	1	NA	04/25/85
NET	C-NET Software for CNI (Cromix only)	2	NA	03/20/84
PAS-D	**68000 Pascal Compiler**	**6**	**02.42**	**08/16/85**
PAS-X	68000 Pascal Compiler	1	02.20	09/17/84
PC Works	**IBM PC to UNIX Communications**	**1**	**NA**	**NA**
Q-PLUS-X	**Q-Office, Office Automation and Wordprocessing**	**2**	**01.94**	**01/30/86**
RBTE	Remote Batch Terminal Emulator	5	01.08	11/17/83
RMCOB-X	68000 RMCOBOL Compiler	1	NA	10/31/84
RMCOBRT-X	**68000 Run-Time RMCOBOL**	**1**	**NA**	**11/07/85**
SGS	SDI Graphics Software	6	02.00	09/06/83
SLMR	SlideMaster Graphics Editor	4	02.03	04/26/83
SMCBAS-X	SMC BASIC Interpreter	1	NA	11/06/84
SMDS	SMD Drive Controller Software	2	NA	07/06/84
SMDS-X	**SMD Drive Controller Software**	**2**	**NA**	**01/27/86**
SPICE-D	SPICE Circuit Design Software	1	11.02	06/21/84
SPMR	SpellMaster Spell Proofing Program	5	01.20	06/27/83
SQL-X	**Informix SQL**	**1**	**NA**	**NA**
STB	**Structured BASIC Interpreter**	**11**	**03.65**	**03/31/83**
STB-D	**68000 Structured BASIC Interpreter**	**1**	**1.00**	**09/24/84**
STMR	**StatMaster Statistical Analysis Package**	**2**	**01.04**	**07/21/83**
TEMR	**TeleMaster Communication Software (C10 only)**	**2**	**02.06**	**04/24/84**
TSDI	**Tri-SDI Graphics Software**	**3**	**NA**	**04/01/83**
UDST-X	**UNIX Documentation Software Tools**	**3**	**NA**	**01/21/86**
UFY-X	Unify Relational Database System	1	NA	11/01/84
ULTCALC-X	UltraCALC Spreadsheet Package	1	NA	12/15/84
UNIX-X	**UNIX System V Operating System**	**3.0**	**NA**	**01/16/86**
UPST-X	**UNIX Programmers Software Tools**	**3**	**NA**	**01/21/86**
WPS	Word Processing Software	11	06.00	04/01/83
WRMR	WriteMaster Wordprocessing Program	10	02.02	11/01/84
X25-D	**X.25 Communications Network**	**1**	**02.04**	**08/06/85**



INSIDE CROMIX

INSIDE CROMIX is an open forum on both eight-bit and 16-bit versions of Cromix. The subject matter is directed towards helping Cromix users derive more from their systems. Members' contributions are invited. INSIDE CROMIX is edited by Jordan Siedband, who can be reached at 5017 Fairview Lane, Skokie, IL 60077, (312)674-1175.

Dear Cromemco,

I have been a good boy for the last year, not even once acknowledging the existence of the IBM-PC or Apple machines. I read the Cromemco manuals, even those parts which were totally unreadable. I even improved my Cromemco with Cromemco parts, of course.

Now, I would like to give you a partial wish list in the fervent hope that the preceding statements will cause you to relent and grant me my wishes. I feel like the person who says, "Lord, grant me Patience, and I Want It Now!"

1. Having tasted UNIX and written shell programs in both operating systems, I would request the availabilty of several shell variables in Cromix so that in a simple command file, $a = #1 assigns the parameter #1 to a shell variable like $a. Later in the program, I can then use the command file copyem.cmd to copy files to the /usr/archive directory by typing:

   copyem /usr/archive file1 file2

   Contents of copyem.cmd:

   ```
   $a = #1
   %loop
   copy -fv #2 $a
   shift
   if .#2 != . go loop
   ```

2. PLEASE!! write ALL new system programs using termcaps.

3. A decent 68000 Cromix spreadsheet!!!

4. A complete termcap (or termcaps) library and routines for use in shell programs and 'C' programs.

5. A 68000 'C' library with math and utilities like in UNIX, with proper rounding and truncating. Because 68000 'C' computes using binary arithmetic instead of BCD like Z80 'C', financial clients scream and moan since pennies do not round well!! SVS's library is, frankly, the pits.

6. Please keep my options open to request more goodies. I promise to continue being as good as I can be!!

A Note About The New Editor, CE

After spending half the day, trying to get CE to work in a multi-user environment and getting INSUFFICIENT TERMINAL CAPABILITIES on my screen, I called Bob Beauchamp at Cromemco who solved my problem. It seems that any termcaps file or ttys file MUST be edited using the -n option of screen. Termcaps and ttys do **not** want to look at **carriage returns**. Also, if you have UNIX or INFORMIX or 68000 COBOL, you already have termcap files. Study those files and be sure that you add those lines that termcaps needs (see help termcaps and help ce).

More About Command Files

I have both Cromemco terminals and AMPEX terminals: AMPEX for me, C-10 for my clients. I like to use screen enhancements for my command files such as highlighting, double-sized characters, etc. Until termcaps can be used with command files, I can use Version 31.05 to build command files for the proper terminal by reading the term attribute. Example: to compile and link a sequence of 'C' programs: (^[, the escape character is entered by pressing \ESC key).

```
%start
term   strcmp -e d-80
% previous line checks to see if the
% terminal is an AMPEX d-80
if -err go p1
echo "^[MHCompiling #1"
echo "^[MK"
go routine
%p1
term | strcmp -e C-10
% check to see if the terminal is a
% CROMEMCO C-10
if -err go p2
echo " ^[dPCompiling #1^[d@"
go routine
%p2
% Generic Terminal
echo " ****** COMPILING #1 ******"
%routine
c #1 |* tee #1.err
if -err go next
code #1.i
if -err go next
crolinker #1 /usr/lib/clib /usr/lib/syslib
    /usr/lib/paslib
del #1.obj #1.err
%next
shift
if #1. != . goto start
```

For the Ampex, the echo displays in double-sized characters; the C-10 shows in reverse field, and if it is neither, a simple message displays. Add more or less to suit your configuration.

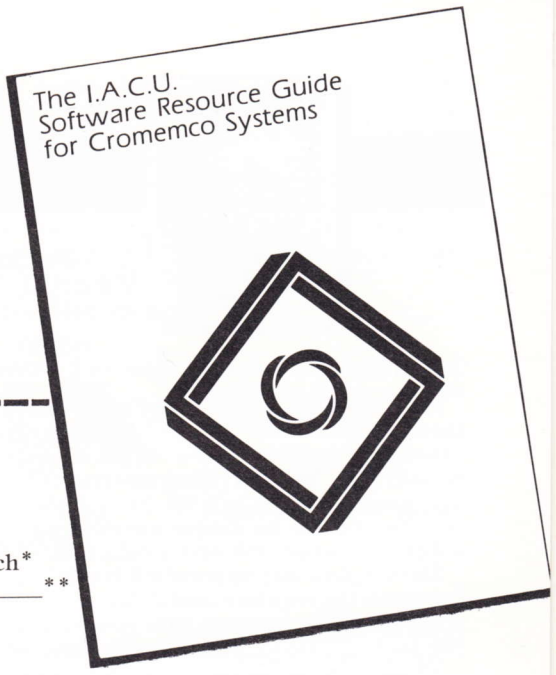
Notice the use of strcmp instead of testinp. The speed is phenomenal!!

Since this is my second column of my new tenure as editor of this column, I would like to thank all of those of you who took the time to respond to the last column. Your confidence and applause are just what I needed to be inspired. Some people think that their "secrets" are superb and no one else can come near them. When you have been programming (applications and systems) for a while, as I have, you realize that anything you can do can be done by someone else, perhaps better. The purpose of a column like this is help each of us improve. Please submit your ideas. Others may be inspired to solve their own computer problems more efficiently.

Having problems locating software?

Now there's help ...

The IACU Software Resource Guide for Cromemco Systems



☆ Over 180 pages

☆ Multiple indices – by product name, application group, operating system, etc.

☆ Complete product descriptions – from source language to technical support

Order your copy now! Only $50

(California residents add 6% sales tax)

The I.A.C.U.
P.O. Box 17658
Irvine, CA 92713
U.S.A.

Send __ copies @ $50.00 each*
I enclose $_____**
☐ Check ☐ Money Order
(make payable to The IACU)

Charge my ☐ MasterCard ☐ Visa
Phone orders with credit card call (714) 661-9764

Card No._____ Expires_____

Signature_____

(Please Print)
Name _____

Address _____

_____

City _____ State _____ Zip _____ Phone _____

Country_____

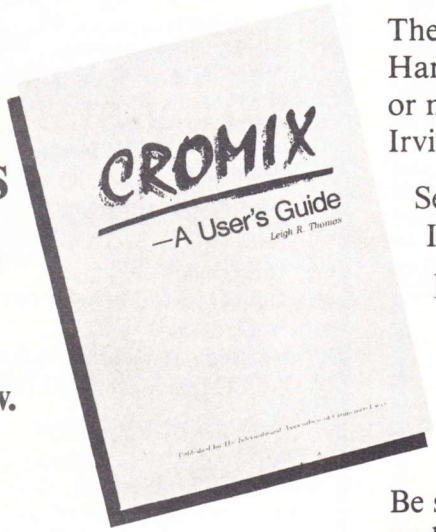
* Calif. residents add 6% sales tax
** U.S. Dollars. International checks not drawn on U.S. bank add $10

Attention CROMIX Users

The book you've been expecting is finally here.

CROMIX — A User's Guide can be ordered now.

(While They Last)



Leigh Thomas' excellent tutorial on CROMIX, reprinted in the U.S., can now be ordered directly from I/O News.

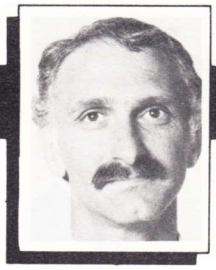
The price is $25.00 plus $4.95 Shipping & Handling per copy. For bulk orders of 10 or more, the price is $17.75 per copy F.O.B. Irvine, California. California residents add 6% sales tax

Send Check, Money Order, or Credit Card Information to:

I/O News
ATTN: CROMIX — A User's Guide
P.O. Box 17658
Irvine, CA 92713
U.S.A.

Be sure to include Membership Number and full shipping instructions with order. (Amounts noted are in U.S. dollars drawn on U.S. banks, only.)

This is a MUST BOOK for any CROMIX user.



C-10 ENCOUNTERS

C-10 ENCOUNTERS is a regular column directed to users of Cromemco's personal computer, the C-10. It is edited by Dr. Tom Beer, of Applied Environmetrics, located at 118 Gordon St., Balwyn, Victoria 3103, Australia. Dr. Beer can be reached by phone during business hours at (03) 817-2571. Submit editorial directly to Dr. Beer.

In a recent issue of Crome-Soma, the official publication of the Cromemco Users Group of Australia, Trevor Campbell asked "How do I write a routine to get a program to run at/after a certain time interval (using interrupts I suppose) on a C-10?" The question itself is a bit like "How do I write a novel?", because there is no unique answer and each person will have a different approach and a different style.

The simplest way to accomplish this is to write a program that generates the requisite time delay and then calls up the required program. If the program that you want to run is a Basic saved file then the delay program can itself be in Basic.

```
10 Input"Enter length of required delay ",L
20 For I=1 To L
30 Rem waste time
40 For J=1 To L
50 Rem waste more time
60 Next J
70 Next I
80 Run"NEXTPROG.SAV"
90 End
```

will run the saved Basic program NEXTPROG.SAV after a variable delay that you set by choosing an appropriate number for L. A value of 110 produces a delay of about one minute. Doubling L gives a four minute delay, and so on. For longer delays insert more loops.

If the program that you want to run is a COM file (say NEXTPROG.COM) then an assembly language program is needed, and one needs to be sufficiently familiar with the operating system to be able to set up the **File Control Block** for the new program, read NEXTPROG.COM into memory, and then jump to the memory location at which your new program starts. If you think about it you will see that this is similar to one of the functions of CDOS. It sits there twiddling its thumbs until you type NEXTPROG at which stage it accesses the disk and starts to load the contents of NEXTPROG.COM into memory, beginning at memory address 100H, until the end of the file is reached. The program instructions are then executed in sequence, beginning at 100H. To make the programmer's life a little easier, CDOS provides system call 136 (88H) "link to new program" that will load and start execution at 100H. Alternatively, you can achieve the same effect with a CDOS command file.

The disadvantage of these programs is that the machine sits there, essentially idle, while it executes the delay program. If the intention is to use the computer to run the program NOWPROG, but after exactly fifteen minutes (say), to automatically run NEXTPROG then the task is slightly more difficult. If NOWPROG is a program that you have written yourself then the task can be accomplished fairly easily. What is required is a subroutine that will obtain the time and if more than fifteen minutes have elapsed since the start time (set at the beginning of the program) then NEXTPROG is called and run. The easiest way to obtain the time is with CDOS system call 146 (92H) — provided that CDOS system call 145 (91H) has first been used to set the time. This is easily done in assembly language code:

```
LD      DE,0        ;Sets time = 0
LD      B,0         ;sets secs = 0
LD      C,91H       ;sets call no.
CALL    5           ;does system call
```

Program to set clock = 0

```
LD      C,92H       ;sets call no.
CALL    5           ;returns with
                    ;mins in B reg.
LD      A,0EH       ;put 14 in A
CP      B           ;compare A-B
RET     NC          ;return if no carry
.....               ;Code to run NEXTPROG
```

Subroutine to see if 15 mins. have elapsed.

The Basic programer will have a bit more problem. The clock can be set with the one line Basic command:

```
10 Print Chr$(27);" 000000" :Rem—note the blank before 00...
```

but there is no similar sequence with which to read the clock.

This means that a Basic programmer will have to incorporate the assembly language read clock routine given above into the Basic program by using the USR command. Fortunately the C-10 Structured Basic manual gives a helpful example on the use of the USR command. Because the number of minutes are returned in the B register we need to set up:

```
20 Data %00D1%,%00C1%,%00CD%,%0005%,%0000%
30 Data %0016%,%0000%,%0058%,%00C9%
40 Integer Byte,Index,Minutes
50 For Index = %0103% to %010B%
60 Read Byte
70 Poke Index,Byte
80 Next Index
```

then follow it with the main program which will have frequent calls of the form:

```
200 Gosub Time'check
```

to the following time check routine:

```
990 *Time'check
991 Minutes=Usr(%0103%,0,%0092%)
992 If (Minutes # 15) Then Return
993 Scr
994 Run "NEXTPROG.SAV"
```

So far we have only utilized software routines to calculate the time. Using interrupts, as suggested by Trevor Campbell, requires more knowledge of both the C-10 hardware and software than I possess. It is difficult to use interrupts on the C-10 because CDOS is already making extensive use of interrupts and you have to know exactly what you are doing to avoid hanging your system. If one examines CDOS in low memory then one finds a jump instruction at memory location 38H. In CDOS C2.61 and C2.65 the jump is to memory location FCB6. (The bytes at 38H are C3 B6 FC) and if the code from FCB6 is unravelled it seems to be concerned with setting up the

screen. I hypothesize that screen operations on the C-10 are being done under interrupt mode 1 — which is the Z80 command IM 1 — so that if the system receives an interrupt it jumps to memory location 38H and executes the program sequence found there. Incidentally, this use by the C-10 of memory location 38H is one of the reasons that CP/M software will often not function on the C-10. Vanilla CP/M leaves the space free, and there are a fair few programs around that use memory location 38H for their own purposes.

If my hypothesis is correct, then the C-10 is spending a goodly chunk of its working life perpetually jumping to memory location 38H and executing the sequence of commands to be found there. So why not replace the jump to FCB6 with a jump to a time check routine first? The idea sounds good and this patch may make it possible for you to use an applications program — say Writemaster — but after exactly 15 minutes have it terminate and flash a message on the screen to tell you to pick your daughter up from her ballet lessons.

However, if one is patching an interrupt service routine, then time is rather critical and one does not have the leisure to use a CDOS system call. The time has to be read straight from the memory location in which it is stored. In Version 2 CDOS, after the clock has been set, the hours are stored in ASCII at F9EA and F9EB, there is a colon at F9EC and the minutes are stored at F9ED and F9EE. The easiest way to use the time is probably on a count-down routine. Set the time to 24:00:00 less the requisite time interval, and have your program terminate when the clock reads 00:00:00.

The location of the time check patch is problematic. The obvious place, namely into the bottom of high level CDOS, will not work because every screen display bank switches memory 8000h to C000h to ROM. Thus the patch must go either below 8000H — which seriously depletes program space — or above C000H which is in the middle of high level CDOS.

After long perusal of high level memory I could see no obvious use for locations FC60H TO FCB0H (under CDOS C2.xx) and so I originally patched it there. However, on subsequent reflection it seemed more sensible to try and put the patch in low level CDOS—i.e. below 100h. Now the bytes between memory locations 38h and 5Ch (the default FCB) do not seem to be used. Thus the trick is to patch the program between 38h and 5Ch, ensuring that it fits, and that a jump to FCB6 terminates the whole thing until the requisite time is found. When this happens the return address on the stack is altered to 0, the time check routine is eliminated, and the interrupt prematurely terminated with a RETI command. The screen will then hiccough, and return with the CDOS prompt.

The Program

```
TOLOC    EQU    00038H
;toloc is Address for patch—if you want it elsewhere change TOLOC
START:   LD     HL,MOVE        ;Move the patch program
         LD     DE,TOLOC       ;to address TOLOC
         LD     BC,BOTTOM-
                MOVE           ;Length of program
         LDIR
         JP     BOTTOM         ;Patch in place — done.
; The patch program follows:


MOVE     PUSH   AF
         PUSH   BC             ;save the registers on stack
         LD     BC,(0F9EDH)    ;get minutes from memory
         LD     A,'0'          ;Test for '0'
         CP     C              ;Tens of minutes = 0?
         POP    BC             ;Flags set: restore BC reg
         JR     Z,$+6          ;Jump only if time is up
         POP    AF             ;Restore A register and flags
         JP     0FCB6H         ;like nothing happened
         LD     A,0            ;If we got here, time is up
         LD     (042H),A       ;so eliminate time check test
         LD     (043H),A
         POP    AF             ;restore A register and flags
         POP    HL             ;Get return address for the
         LD     HL,0           ;interrupt routine, replace it
         PUSH   HL             ;with a return to 0
         RETI                  ;Abort the interrupt routine
;The patch program has now ended
BOTTOM:
         JP     0              ;Return to CDOS
         END    START
```

The End

I find it easiest to use the above program with a CDOS command file so that as soon as the requisite time has elapsed and execution returns to CDOS the next program in the command file is executed. This program could, for example, be a large message drawn with EASEL telling you to pick up your daughter from ballet.

Permanent Clock Display

A bit more delving around in high memory reveals that the region around FA40 is used for the screen's 25th line display (generally called message deposit). Thus if one wants to see the clock on all the time after setting it, without having to invoke it from the command line, replace the portion of the above program with the following commands:

```
MOVE     PUSH   AF
         PUSH   BC
         PUSH   DE
         PUSH   HL             ;Registers saved
         LD     HL,0F9F1H      ;Status line clock
         LD     DE,0FA51H      ;25th line display (far right)
         LD     BC,8           ;count
         LDDR
         POP    HL
         POP    DE
         POP    BC
         POP    AF
         JP     0FCB6H         ;orig. jump addr. from rst 38
```

The above programs are unlikely to work with CDOS C3.?? (Release 5 and higher) because all the addresses are likely to be different. A more sophisticated version of the time check patch would check the address at memory locations 39 and 40, and use whatever it found rather than the absolute address

C-10 Encounters

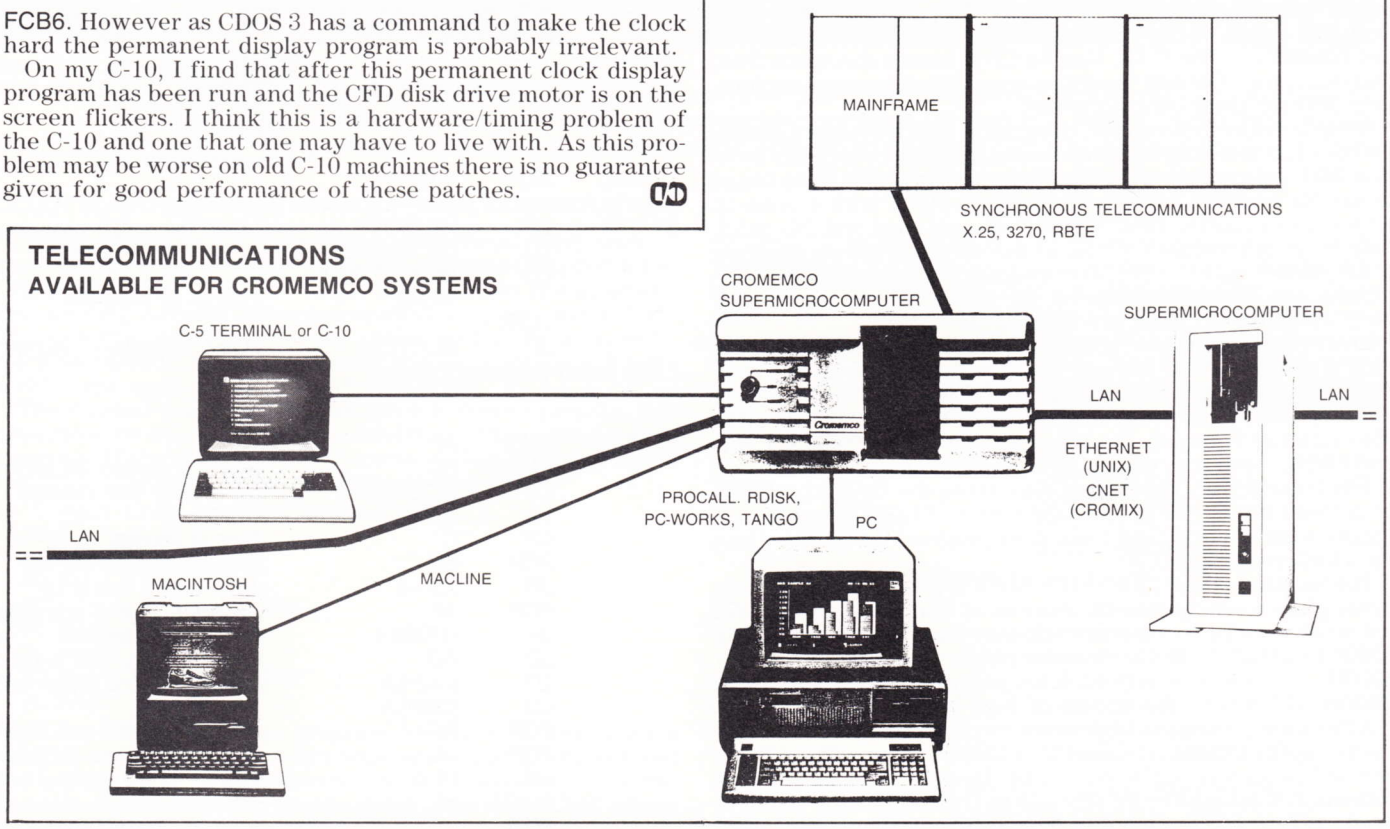
Continued

FCB6. However as CDOS 3 has a command to make the clock hard the permanent display program is probably irrelevant.

On my C-10, I find that after this permanent clock display program has been run and the CFD disk drive motor is on the screen flickers. I think this is a hardware/timing problem of the C-10 and one that one may have to live with. As this problem may be worse on old C-10 machines there is no guarantee given for good performance of these patches.

Telecommunications

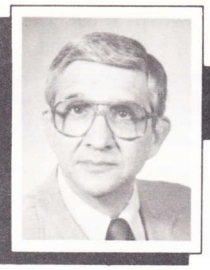
Continued from page 13



SOFTWARE CONSORTIUM

I.A.C.U. member Jon Rosser is engaged in creating a "software consortium," which would be comprised of Cromemco end-users and dealers. Its purpose would be to orchestrate the mass acquisition of software for the group and its distribution; the compilation and cataloguing of software product fact sheets; perform rigorous software evaluations and document the findings; and to influence the direction of new software developments in line with the interests of the group. If you're interested in the idea, contact:

Jon Rosser & Associates
1613-A West Division
Arlington, Texas 76012
(817)261-8620



32K CLASSROOM

32K CLASSROOM is a regular column aimed at explaining programming techniques using Cromemco Structured BASIC. It is edited by Bernie Thomas, President of Jakes Manufacturing Corp., P.O. Box 23050, Nashville, TN 37202. Submit any editorial contributions to I/O NEWS in care of the 32K CLASSROOM.

One of the really nice features of Cromemco's 32K and 68000 Structured Basic is the LIBRARY. To demonstrate the use of LIBRARY, CALL, BEGINCOMMON, ENDCOMMON, and ENDPROC, I have chosen two routines which I use to reduce the length of strings in order to save on file storage space. The first of these, .strip, strips the spaces out of a string. The second, .rebuild, replaces the spaces in the string. I have shortened these routines here since this is really a demonstration of the LIBRARY command. The actual routines handle such names as "DePaul", "McHenry", "MacDonald", and addresses such as "106 101st Street", etc., and eliminates user nuisances such as leading and trailing spaces. If you should have an interest in seeing the full routines, please let me know. I also use one called .abbreviate, which makes certain standard abbreviations.

The first thing you must do in order to use LIBRARY and CALL is to write the routines which you will add to a LIBRARY and SAVE them to the disk. The following is an example of two PROCEDUREs, or routines, which we will name .strip and .rebuild.

```
  10 Procedure .Strip
  20   Begincommon
  30   Dim S$(49)
  40   Endcommon
  50   Dim Sub$(49)
  70 *Reduce'to'lower'case
  80     For By=0 To Len(S$)-1
  90     If S$(By,By)>="A" And S$(By,By)<="Z" Then  Do
 100       S$(By,By)=Chr$(Asc(S$(By,By))+32)
 110       Enddo
 120     Next By
 130 *Establish'caps
 140   P=Pos(S$," ",P+1)
 150   If P=-1 Or P=Len(S$)-1 Then P=0 : Goto Strip'spaces
 160   If S$(P+1,P+1)>="a" And S$(P+1,P+1)<="z" Then  Do
 170     S$(P+1,P+1)=Chr$(Asc(S$(P+1,P+1))-32)
 180     Enddo
 190   Goto Establish'caps
 200 *Strip'spaces
 210   P=Pos(S$," ",P+1)
 220   If P=-1 Or P=Len(S$)-1 Then Goto Cap'first'byte
 230   E=Len(S$)-1 : Sub$=S$(0,P-1)+S$(P+1,E) : S$=Sub$
 240   Goto Strip'spaces
 250 *Cap'first'byte
 260   If S$(0,0)>="a" And S$(0,0)<="z" Then  Do
 270     S$(0,0)=Chr$(Asc(S$(0,0))-32)
 280     Enddo
 290   Endproc
5000 *Sav : Save"Strip.pro"
```

```
  10 Procedure .Rebuild
  20   Begincommon
  30   Dim S$(49)
  40   Endcommon
  50   Dim Sub$(49)
  60     For By=1 To Len(S$)-1
  70     If S$(By,By)>="A" And S$(By,By)<="Z" Then  Do
  80       E=Len(S$)-1 : Sub$=S$(0,By-1)+" "+S$(By,E) : S$=Sub$
  85       By=By+1
  90       Enddo
 100     Next By
 110   Endproc
5000 *Sav : Save"rebuild.pro"
```

Every Library Procedure must begin with the command Procedure followed by the name — which *must* begin with a period — as in line 10.

Since we are going to pass a string from the calling program to our procedure and back to the program, we must have a commonly dimensioned string variable which we will call S$. In a Procedure, this common variable must be preceded by Begincommon and followed by Endcommon. In a calling program the Begincommon is understood, and therefore not necessary. The Endcommon is necessary in both cases, however.

The Procedure is terminated by Endproc, which returns control to the calling program.

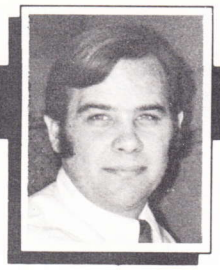
After you have written your Procedures and SAVEd them to the disk, run "libbuild.sav." Answer C, for Create, to the first prompt. For this demonstration, name the Library strings.lib. After your Library has been created, and you are prompted again, answer A, to Add your Procedures to the Library, and answer strip.pro when asked for the Saved Procedure Name. When you are returned to the Menu, answer A again, and answer rebuild.pro . After you are returned to the menu, answer Q to Quit. To try out the Library strings.lib run the following program.

```
 100   Library"Strings.lib"
 110   Dim S$(49)
 120   Endcommon
 130 *Ask : Inout"Enter your string )",S$
 135   If S$="end" Then Stop
 140   Call .Strip
 150   @ : @ S$
 160   Call .Rebuild
 170   @ : @ S$
 180   @ : Goto Ask
```

Line 100 uses the Library command which opens strings.lib. If you are going to use more than one Library in a program, Library and the name must be given immediately preceding the Call. The Library statement does not have to be the first line of a program. As with Kopen, however, it must be the last command on a line. Line 120 uses the Endcommon command as mentioned above, and as I said, the Begincommon is not necessary in this program. The Call command is used in Lines 140 and 160 in the same manner as you might use a Gosub.

I use .strip and .rebuild in hundreds of programs, and if I had to include the code for .strip and .rebuild in every program, it would consume lots of time and disk space.

If you are not presently using Library, I strongly suggest you give it a try. Good Luck, and please contact me if you have trouble or wish further information.



TEC TIPS

TEC TIPS is a regular column aimed at providing hints for keeping systems up and running. It will not attempt to deal with specific engineering applications or non-standard configurations. TEC TIPS is edited by Richard Quinn, owner of QUINN TEAM, a Southern California computer service firm. Telephone (818) 889-4819

FTAR Check Sum Errors

I was recently using ftar and constantly got "check sum" errors on known good disks that seemed to happen at random until I noticed that they occurred when flush was running. I don't know why flush causes this error, but sure enough, when I stopped flush, using kill -3 and sync, the problems went away. This leads me to believe that there are likely errors induced when using ftar to write disks (although I don't know this for sure). My guess is that flush interferes with the timing of the FDC card. I have not tested this problem on all FDCs even though the problem I encountered was highly repeatable using a 16FDC and Cromix Plus version 31.05.

I therefore recommend killing flush when doing tar or ftar reads and writes. In fact, I know that flush does not always clear STDC buffers but only RAM memory buffers. Tests I have run (writing a file to an STDC drive and then exiting and turning off the system) prove that the file never is written to the disk. On the other hand, kill -2 1, which shuts down the system will always force a flush of the STDC buffers to the drive causing those files to be written and closed on all mounted and root devices. On Cromix-Plus systems (version 31.05) the sync utility should be run after flush is killed to ensure that all STDC and RAM memory buffers are written to the disk.

CTD—Cartridge Tape Drive Information

Let me start this discussion with a short description of the CTD tape unit and how it is interfaced to the Cromemco.

The CTD is built around the **Cypher** floppy tape drive. This is a revolutionary tape drive that is built with a microprocessor controller to look like a floppy disk drive and yet use a tape so that the amount of information stored is greatly increased. It therefore interfaces to the computer through the floppy disk controller, the 64FDC.

The interface is slightly different in that the tape contains 6 streams of data on one quarter-inch-wide tape. A stream is a single track of information and goes from the beginning of the tape to the end, with the next stream starting at the end and going back to the beginning of the tape. If you were able to split the tape into six narrow bands and lay them end to beginning you can see that you would have a very long recording track in a smaller package.

Anyway, the interface is different as it uses two disk drive select lines and the side select line to control the drive rather than just one drive select line, thus giving selection of all the various streams. Since the tape drive uses two drive select lines, this means that only two floppy drives can be used on a system with a CTD. That is why there are two drivers, /dev/ftab and /dev/ftcd. Device /dev/ftab uses the A and B drive select lines and /dev/ftcd uses the C and D select lines. Using device /dev/ftcd is the norm for Cromemco.

There are several problems with tape over floppy when it comes to recording data, but this ability to put a lot in a small package is the primary reason that tape is needed for back-up. You just simply can not put any where near as much on floppies without lots of disks and lots of time. But, from a technical standpoint, there are some problems with tape which make it harder to use.

The first problem is starting and stopping the tape. Unlike a floppy which is going over the same physical track over and over (as long as the heads are not stepped), a tape must be stopped, started, and repositioned. If you don't pick up the information on the first pass, the drive has to be stopped, backed up and run forward again. If it slips, the data may be read or written in a slightly different spot. Also, the tape reels are heavier, making the drive transport bigger and thus hurting the ability to stop and start the tape due to weight.

If that isn't enough, the tape can stretch and cause the data cells recorded to expand and become unreadable. Also the heads have to be able to move to each of the 6 tracks and may suffer accuracy, especially if the tape does not track up or down in a perfectly straight line. With all of these problems you wonder how they ever made them work — a tribute to modern engineering!

The way a tape works best is in the "streaming" mode, that is, data is presented to the drive fast enough to prevent the drive from stopping, backing up and restarting. Cromemco uses a method whereby the data to be written is first loaded into memory and them dumped in a batch to the tape drive, thus keeping the start/stops to a minimum. But the data transfer rate of the floppy type interface is just too slow to prevent stop/starts.

Under ALL CIRCUMSTANCES be certain that flush is not running. This is important in that the occasional flushes will change the timing of inittape and reads/writes to the tape.

The more recent utilities (inittape) under Cromix-Plus tells you where the bad sectors are and creates a "bad block" list so that the tape will avoid those areas when reading and writing data. It does this by reading (verifying) the tape through from 1 to 10 read alls. The bad sectors are listed and avoided whenever the tape is used, much like an alternate track table on a hard disk.

With that background, let's move on to current issues. We began to notice that certain tapes had large numbers of bad areas, all in the same spot. We decided to spend time and effort to find the source (software, tape drive, or tape cassette) of the problems. Some interesting details came out.

First, we had a long conversation with Cypher — the folks that make the tape drive. They told us that Scotch, the major maker of tapes, had produced a large number of defective tapes. You can tell by looking at the date code on the bottom metal plate of the tape. There is a five digit number followed by a 4 digit number in the form of XXXXX XXXX. If you skip the first two digits the next one is the year followed by the week the tape was made. Example: The code date 58516 1855 was made in 1985, the 16th week; XXYWW XXXX where Y is the year and WW is the week.

Tapes made from the 15th to the 37th week of 1985 are having MANY problems. Talking with Scotch, we were unable to get a clear picture of why this is so. They are admitting that there is a problem but will not go so far as to say the tapes are defective. Also, they don't seem to want to tell us what the difference is between these tapes and others. But we have traced our most unreliable tapes in ALL CASES to tapes made during this period. Cypher says the tapes are defective and it is not their drive. Scotch says they will replace the tapes found to be defective but have not said how we are to receive credits. We hope to find out and let folks know.

I have never been fond of 3M's (Scotch) magnetic media products. I found them to be very poor in consistency and the diskettes are abrasive to drive heads. Many have even caused damage to drives I have repaired, but

mostly the disks would develop read/write errors young in life.

We are seeing other makers of the DC-600 type tape cartridges and this will help 3M get their act together, or at least give us a new source for tapes. Also, like disks, volume helps quality improve.

Now that you have all this background — maybe more than you wanted — here are some tips to make your CTD work better. Before you back up or restore from a tape, open and close the door on the tape drive several times to "re-tension" the tape. This runs the tape all the way forward and back to take up slack. This is important on new tapes before their first use. Do this two or three times.

Make certain that you use the most current inittape utility. This can verify the tape and identify bad sectors. Use inittape after every few uses of the tape to recheck it if you suspect problems are developing. (Some tapes are real troopers and hold up well while others degrade with use. Knowing your tape's "personality" will give you a feel for how often to check it.)

For backup, I use ftar. It is fast, reliable and easy to use. I use command files to do regular backup with the name of tapeback.cmd. Also, when backing up the disk, I get into the directory to be backed up and refer to it as " " (meaning the current directory). For example, the command ftar -vc /dev/ftcd . will back up the directory you are in with all descendants displaying the names as it goes. If you are in the root directory (/) when executed, ftar will back up the entire disk — provided you don't exceed the tape's capacity (about 25megs)

The advantage of this syntax is that the file name is stored with the ftar achieve as specified. For example if I say ftar -vc /dev/ftcd /, the full path name of screen.bin is /bin/screen.bin. That means when I do a restore from my ftar device, ftar will put screen.bin in the /bin directory no matter where I may want it. If on the other hand I have used the period "." meaning the current directory, and I'm in the root directory when backing up screen.bin, the path name stored by ftar is ./bin/screen.bin which means I can restore to any directory because the restored tree will build from the directory where I am in the system, not necessarily where the file came from.

If I restore from the root directory — the same directory I backed up from -- the path name will be the same. But if I want to restore my root device tape to another segment of the drive or another directory to keep from overwriting the same file names still on the disk, all I have to do is mount the other segment, get into that directory, and restore to it rather than the original path directory.

If the ftar names have been stored using absolute path names, they will be restored absolutely. But if I use relative path names, I can restore them from where I am in the file system, allowing options for file restoration that will be lost with absolute names. I hope that's clear (if not, ask Bill.)

Improving CTD Operation by Checking the 64FDC

In terms of hardware, there are several things you can do to improve CTD tape operations. First, be certain that the 64FDC is set up according to the system in use. These set-ups have changed in recent weeks and are different for XPU and DPU systems and are also 64FDC revision dependent. These changes are VERY important and a MUST for any tape drive. The most recent information is available from Cromemco dealers with access to the Cromemco Technical Bulletin Board.

The adjustment of the phased locked loop on the FDC — that little adjustable capacitor in the upper left hand edge of the card — is very important. This should be adjusted to the tape drive and will then be fine with disks. It adjusts the free running frequency of the read circuit data separator and is adjusted higher or lower as it reads the tape or disk to match the frequency that was recorded on the tape or disk. This is one of the great features of this state of the art controller.

But if the free running frequency is too far off, it will not "lock on" to the tape fast enough to prevent lost data, and since the tape does not come around again like a disk, a CRC error will be reported.

Cromemco sets this adjustment at the factory using a frequency counter. I use a scope watching pin 6 of IC-1 to "see" how much the phase locked loop shifts when reading. This will be hard for you to do if you don't understand what to look for on the scope, and without knowing you will most certainly foul up your controller. The phase locked loop affects read operations on the drive only and will not affect writes.

Properly adjusted, the controller will lock on fast, read a wide data window assuring that all data in a sector is received and improve greatly the reliability of the drive. I have seen this simple technique take the drive from no go at all to flawless operation.

Soup Campbell in Anchorage, Alaska has spent much time and many phone calls with me to prove this change and I thank him for that time and effort.

New RDOS Available for STDC Machines

There is a great new RDOS for STDC drive machines. It is **version 3.12** and will allow boots directly to an STDC hard disk drive. It can be used in 64FDCs and properly set-up 16FDCs. If you are currently using a 64FDC, or have RDOS 3.08 in your 16FDC, you can use this by simply changing the ROM in the card.

The recommended versions of RDOS (other than C-10s) are as follows:

• All CDOS machines and Cromix machines using floppies only use version 2.01 or 2.52. 2.52 is best as 2.01 has a problem when trying to do disk drive alignment options. Also, 2.52 allows you to boot to any floppy drive which can be a life saver if your "A" drive goes on the blink.

• Cromix-Z-80, Cromix-Plus, and Cromix-D machines using the WDI and WDI-2 hard disk drives use RDOS 3.08. 3.08 allows for boot directly to the WDI hard disk and also allows testing of all banks of installed memory.

• The same systems as above, but using the STDC type hard disk drives are best if equipped with the newest version, 3.12. This gives the same memory test functions and allows direct boot to STDC drives.

In all cases, a boot must be written to the Cromix drive to work (CDOS does not have any direct-to-hard-disk boot options). For the WDI type drives use wboot hd0 in 11.27 Cromix, Cromix-D, or Cromix Plus version 31.05 or newer. (Less than 31.05 in Cromix-Plus did not support WDI drives.)

For STDC type drives, use wboot std31 to write the boot. Be certain to use std31 as this refers to the whole drive and not a segment.

Disk drive alignment options are only in RDOSs 2.01 and 2.52. RDOS 3.08 and 3.12 have no disk alignment options but will test their respective hard disk types and will provide expanded memory testing.

All versions of RDOS will allow boots to any floppy drive except version 2.01 which will boot to floppy drive "A" only. The RDOS B for **boot** command will always boot to the default boot device as determined by the switch settings on the FDC, which could be any of the floppies or hard disk drives supported by that particular RDOS. By entering the b followed by the device designator any drive can be made to boot regardless of the default device. On power up, hit ESC to stop the auto boot and enter ba for boot to drive A, bb for boot to drive B, etc. Booting to WDI type drives (RDOS 3.08) is bh0; to boot STDC drives (RDOS 3.12) enter bst0.

I have found the use of the proper RDOS greatly simplifies computer startups and I highly recommend them.

DISKINFO Utility Very Handy

I find the new diskinfo utility to be very handy in knowing how to set up a hard disk after a system problem. The problem is that most people don't use it or even know about it until there is a problem which may too late.

When you have changed something on the hard disk, or if you have not already done so, use diskinfo to print out a list of all partitions and alternate tracks on the hard disk(s) for your system so that you will know what were the original setups.

Typing diskinfo std31 for instance will tell you all about the first STDC type drive on the system. If the output is directed to a printer (diskinfo std31 V /dev/prt) then a hard copy of the disk parameters will be printed. This will aid technicians and operators alike if the drive has to be re-initialized at anytime in the future.

SOFT TIPS

SOFT TIPS is a regular column aimed at providing software oriented hints and ideas. Member contributions are encouraged. SOFT TIPS is edited by Norman Vadnais, President of Computer Specialists & Associates, an Orange County Customer Support Specialist. Mr. Vadnais can be reached at (714) 841-3620.

The following programming technique was submitted by Robert Brown, *of* EXCALIBUR COMPUTERS, *in response to an inquiry regarding aging routines. Due to its general applicability, the response is reproduced here. Any questions or comments may be directed to Mr. Brown at:*

EXCALIBUR COMPUTERS
4548 Auburn Blvd., Suite 191
Sacramento, CA 95841
(916) 971-9610

AN AGING ROUTINE

In order to do aging, it is important to be able to compute the number of days between two dates. I do this by converting each date to be a **julian day** and taking the difference. I use a procedure to compute the julian day. My procedures are all in a library; they can also be imbedded in the program.

Mo = Integer, Month of the date in question
Dy = Integer, Day of the date in question
Yr = Integer, Year of the day in question
Jday = The julian day. Given two julian days, you can take their difference and that would give you the number of days between the two dates.

```
10 Procedure .To'julian (Mo,Dy,Yr)
20   If Mo>2 Then Mo=Mo+1 : Goto 40
30   If Mo=1 Or Mo=2 Then Yr=Yr-1 : Mo=Mo+13
40   Jday=Int(365.25*Yr)+Int(30.6001*Mo)+Dy
50   Endproc (Jday)
```

Take my word for it, it works and is a lot faster than the brute force method. I found the algorithm in one of my old astronomy books on calendars. The range of dates for which this routine produces valid results is 1935 to 2099.

You can also get a julian day, add a number of days to it, get a new julian day and convert it back to the date. The procedure for that is as follows.

```
 10 Procedure .From'julian (Jday)
 20   Yr=Int((Jday-122.1)/365.25)
 30   Mo=Int((Jday-Int(365.25*Yr))/30.6001)
 40   Dy=Jday-Int(365.25*Yr)-Int(30.6001*Mo)
 50   If Mo<14 Then  Do
 60     Mo=Mo-1
 70     Else
 80     Mo=Mo-13
 90     Enddo
100   If Mo=1 Or Mo=2 Then Yr=Yr+1
110   Endproc (Mo,Dy,Yr)
```

Also, in my programs the year is stored as a number from 1 to 99. The following procedure does the necessary conversion for me in the julian routines.

```
10 Procedure .Convert'yr (Yr)
20   If Yr>50 Then Yr=Yr+1900
30   If Yr<=50 Then Yr=Yr+2000
40   Endproc (Yr)
```

To use these in an Aging Report, here is a sample subroutine:

D'cur'chg$ = Date of current Charge stored in MM/DD/YY format as an ASCII string.
Cur'julian = An integer that has already been defined as the julian day for today in a previous routine.
Difference = An integer which is the number of days past due an account is. If is negative, then it is the number of days till the account is due.

```
2090 *Compute'age
2110     Mo=Val(D'cur'chg$(0,1))
2120     Dy=Val(D'cur'chg$(3,4))
2130     Yr=Val(D'cur'chg$(6,7))
2150     .Convert'yr (Yr;Yr)
2160     .To'julian (Mo,Dy,Yr;Julian1)
2170     Difference=Cur'julian-Julian1
2180     If Difference<30 Then Gosub current'account'routine
2190     If Difference>=30 And Difference<59 Then Gosub thirty'day'routine
2200     If Difference>=60 And Difference<90 Then Gosub sixty'day'routine
2210     If Difference>90 Then Gosub ninety'day'routine
2350   Return
```

For those who prefer 'C', here are the functions I use in that language.

```
to_julian (date_buf)
char date_buf[];

{
 int  mo,dy,yr;
 long jday = 0;
 yr = date_buf[2] + 1900;
 mo = date_buf[0];
 dy = date_buf[1];

/* rountine should be good from 1935 to 2099 appx   */
/* the base year is 2020                            */
 if (mo > 2)
    mo += 1;
 else
   {
    if (mo ==1 || mo == 2)
         {

          yr -= 1;
          mo += 13;
         }
   }

 jday =  (double)(365.25 * yr);
 jday += (double)(30.6001 * mo);
 jday += (double)(dy - 735785L);

 if (jday > 32766 || jday < -32766)
   syserr("To Julian:  Value out of bounds");

 return ((int)jday);

}
/*****************************************************************/


from_julian (inp_jday,date_buf)
int inp_jday;
char *date_buf;
{
 long mo,yr,dy;
 long temp,temp1;
 long jday;
 jday = (long)inp_jday + 735785L;

 dy = jday;
 yr = (dy - 122.1)/365.25;
 temp = 365.25*yr;
 mo = (dy - temp)/30.6001;
 temp1 = 30.6001*mo;
 dy = dy - temp - temp1;


 if (mo < 14)
    mo -= 1;
 else
    mo -= 13;

 if (mo == 1 || mo == 2)
    yr += 1;

 yr -= 1900;
```

```
 *(date_buf+0) = (char) mo;
 *(date_buf+1) = (char) dy;
 *(date_buf+2) = (char) yr;

}

/*****************************************************************/


dy_of_wk (inp_jday)
int inp_jday;

{
 int day;
 long jday;
 jday = (long)inp_jday;
 jday += 735785L;

 day =  (jday + 5) % 7 + 1;

 return(day);
}


time (time_buf)
char time_buf[];

{
    printf("%02d:%02d:02d",time_buf[0],time_buf[1],time_buf[2]);
}


date (date_buf)
char date_buf[];

{
int yr=  date_buf[2]+1900;
static char *month_name[13] = {
                                0, "Jan", "Feb", "Mar",
                                   "Apr", "May", "Jun",
                                   "Jul", "Aug", "Sep",
                                   "Oct", "Nov", "Dec"
                               };

printf("%s-",month_name[date_buf[0]]);

printf("%02d-%d",date_buf[1],yr);

}

day (day_c)
char day_c;

{
 static char *day_name[8] = {
                               0,"Sun",
                                 "Mon","Tue","Wed",
                                 "Thu","Fri","Sat"
                             };

printf("%s",day_name[day_c]);

}



stime (time_buf, time_str)



char time_buf[];
char time_str[];
```

Continued

Soft Tips *Continued*

```
{
        sprintf(&time_str[j++],
                "%02d:%02d:%02d",time_buf[0],time_buf[1],time_buf[2]);
}


sdate (date_buf,date_str)
char date_buf[];
char date_str[];

{
int i;
int yr=  date_buf[2]+1900;
static char *month_name[13] = {
                                0, "Jan", "Feb", "Mar",
                                   "Apr", "May", "Jun",
                                   "Jul", "Aug", "Sep",
                                   "Oct", "Nov", "Dec"
                              };

sprintf(&date_str[0],"%s-",month_name[date_buf[0]]);

sprintf(&date_str[i++],"%02d-%d",date_buf[1],yr);
date_str[11] = NULL;

}

sday (day_c, day_str)
char day_c;
char day_str[];

{
 static char *day_name[8] = {
                               0,"Sun",
                                 "Mon","Tue","Wed",
                                 "Thu","Fri","Sat"
                             };

sprintf(day_str, "%s",day_name[day_c]);
day_str[3] = NULL;

}
```


COMMERCIAL MEMBER
Western United States

MULTI-MEDIA VIDEO INC.
3350 Scott Blvd., Bldg. 21
Santa Clara, CA 95051
Tel: (408) 727-1733
Tlx: 171-577 MMV USA

Multi-Media Video, (MVV), markets bilingual Arabic/English Cromemco systems and peripherals throughout the Middle East. Installations have been made in the government and banking sectors; a complete Arabic banking system was developed for the latter.

Key Personnel: A.B. Kader, President
Miguel Mora, Sales Manager
Jill Peterson, Marketing Manager

Major Market Area: Authorized dealers in Egypt, Saudi Arabia, and Pakistan.

COMMERCIAL MEMBER
Mid United States

COMPUTER CROSSROADS OF AMERICA, INC.
6 Terrace Shopping Center Richardson, Texas 75081
(214) 231-6108 Twx/Telex 4991118

We are a CROMEMCO MASTER DEALER engaged in DEALER and OEM sales, service and support. We are in the top twenty-five dealers in the U.S. We have a consulting staff comprised of specialists in hardware, software and applications engineering. We are presently engaged in sales from the hardware level (equipment and/or software delivered in an unopened box) through the complete systems level where we take full responsibility for the system hardware configuration, instruction and maintenance of a system. As our name implies THIS IS THE CROSSROADS WHERE IT ALL COMES TOGETHER.

Key Personnel: Ed Fearon, President, Sales & Support
John Rateau, Sales & Support
Danney Jarman, Sales & Support
Joe Essler, Sales & Support
Bill Carnahan, Support

Major Market Area: Sales & Service Worldwide

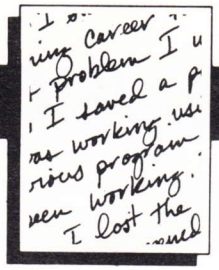
COMMERCIAL MEMBER
Canada

D.E. SYSTEMS LTD.
1284 Wellington St.
Ottawa, Ontario
Canada K1Y 3A9
(613) 729-5164

D.E. Systems Ltd. is a full service company offering Cromemco Hardware, Software Development, Education and Application Programs. We have developed integrated Inventory, Point-of-Sale, Invoicing, Accounting and Sales Analysis programs as well as a Courier Package. We specialize in Cromemco Computers for government and small businesses. We have most Cromemco products in stock and offer technical support on the hardware and software. We offer maintenance of all Cromemco equipment and related peripherals.

Key Personnel: Bruno Dugas, President
Keith Corkum, Director (Systems Development)
Dwight Presley, Senior Analyst

Major Market Area: Eastern Canada



USER NOTES

USER NOTES are useful techniques, tips, ideas and other helpful information gleaned from our member's experiences with their Cromemco systems. If you have something along these lines that you want to share, write it up and send it to I/O NEWS, c/o USER NOTES, PO BOX 17658, IRVINE, CA 92713.

The following notes were contributed by Donald H. Madsen *(I.A.C.U. member #280). Correspondence to Mr. Madsen can be sent to:*

Donald H. Madsen, P.E.
Consulting Engineer
1315 Whiting Avenue Court
Iowa City, Iowa 52240

I am enclosing some brief notes relating to the /etc/termcaps and /usr/help.msg files of Cromix-Plus, version 31.05.

I am very pleased with Cromemco's effort to make the operating system easier to use. The availability of the Cromix-Plus driver package at a reasonable cost was also very much appreciated.

There are some minor problems in the TERMCAPS description in the Cromix-Plus User's Reference Manual (Rev. C). On page 218 the definitions of 'cl' (Clear to end of line) and 'ce' (Clear screen) seem to be interchanged. I think they should be

cl str Clear screen
ce str Clear to end of line

Also it seems that each line in /etc/termcaps must end with a Line Feed and not with a Carriage Return—Line Feed. This can be accomplished with the -n option in screen or the default usage of CE.

The new editor, CE, works fine with my TeleVideo 950 terminals as does the Command Line Editor (a real convenience). My T950 section of termcaps is as follows:

```
# TeleVideo TV 950 Terminal
#
T2¦T950¦TeleVideo 950 Terminal:\
        :co#80:li#24:\
        :up=^K:do=^J:nd=^L:nb=^H:\
        :dl=\ER:al=\EE:\
        :cm=\E=%+ %+ :\
        :cl=\E*:cd=\Ey:ce=\Et:\
        :dc=\EW:ic=\EQ:\
        :kR=^R:kI=^I:kE=\E:\
        :kl=^H:kr=^L:ku=^K:kd=^V:\
```

The TeleVideo 925 included in the Cromemco file has '\E + ' for *clear screen* and I am not sure that is correct, but I do not have a TeleVideo 925 manual. In the TeleVideo 950 '\E + ' is *Clear unprotected to Blank character*.

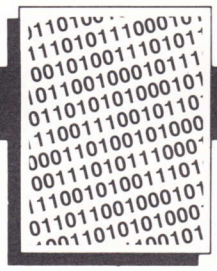
I have also added /usr/help/help3.msg for the TeleVideo 950 and linked it to /usr/help/help.msg in order to make the on-line manual print correctly on the screen.

For TeleVideo 950 /usr/help/help3.msg: (see page 76 in Reference Manual)

```
\033*Directory of topics:\n\n
\nTopic ?
\t\t\t\t\t\t\t\t   \033)%1d%% MORE\033(
\t\t\t\t\t\t\033)PRESS SPACE BAR TO CONTINUE\033(
\t\033)COMMANDS: Begin Help Quit Return Up <space> <up arrow> <return>\03
\r\033t
\033*
\036
\033E
\033=8<SP>
```

In the last line V SP V indicates a space (which is invisible on the screen). V SP V could probably be replaced with \040 to make it visible.

Then: link -f /usr/help/help3.msg /usr/help/help.msg



BITS & BYTES

BITS & BYTES is the place to look for the odd bit of information, opinion, programs, profiles and rumors that circulate through The IACU. Our ears are always attuned to any interesting miscellany — if you have something to contribute send it along to I/O NEWS • P.O. Box 17658 • Irvine, CA 92713 • (714) 661-9764

CE Benchmarks

In the last issue of *I/O NEWS*, we introduced the new 68000-based CE editor for Cromix-Plus (ver. 31.05) and UNIX (release 3). The following benchmarks and comments were recently submitted by Dr. Curt Terwilleger of Cromemco:

Editing Task	ce	screen
1. Do 1000 substitutions of equal-length strings	19.3	29.8
2. Do 1000 substitutions of 9-character string for 3-character string	44.8	194.8
3. Find a given string at end of 112,000 byte file	2.9	52.9
4. Move cursor to end of 112,000 byte file (Jump)	.2	39.6
5. Move cursor to start of 112,000 byte file (Jump)	.2	47.4

Test number 1 shows how 68000-based ce is about 50% faster than Z80-based screen in straightforward character manipulation. The command used was 1000 S /abe/, /ebe/.

Test number 2 shows the dramatic advantage of the line-base internal structure of ce versus the string structure of screen. In this case each line was lengthened by the substitution of 9 characters in place of 3; ce performed the job 4 times faster than the older editor.

The remaining tests show the overwhelming speed improvement which is made possible by keeping the entire text in memory.

In test 3, each editor is asked to find the first occurrence of a string which has been placed at the end of a 112,000 byte file; ce located it a full 50 seconds faster than screen.

Tests 4 and 5 show that ce can move the cursor around a large file in 0.2 sec, or one-tenth the time it takes to re-write a single screen of text at 9600 baud. The older screen program takes about 40 seconds to do the same job.

These benefits of ce come at a price, albeit a very reasonable one: it takes longer to load or save a file than it would under screen. For example, the 112,000 byte file mentioned above can be read into screen in 6.9 seconds (since only a fraction of the file is read initially), while ce requires 17.8 seconds. Similarly, the time to update that file is 5.6 seconds for screen, and 11.0 seconds for ce. This time is quickly made up, of course, by the fast operation of ce during actual editing.

Notes on Cromix-Plus ver. 30.79

I experienced some problems with Cromix-Plus version 30.79. Most of the problems occurred when there was a lot of disk I/O. It kept messing up the files in /etc, and I had to resort to using 30.51 until recently when I replaced IC 38, (part – 502-0123) on the STDC (revision C) with a newer version (part – 502-0123-1). After that the system seemed to work fine. I don't know if other users had a similar problem, or if this was only an idiosyncracy in our system. Likewise, the newer version of Cromix-Plus (ver. 31.05) may also correct it. Nonetheless, this info may be of some use to others.

Paul Lee, Instructor
Physics and Technology Dept.
Canadian Union College
College Heights
Alberta, CANADA TOC OZO
1-403-782-3381 ext. 222

IACU Chess Players?

I am presently in the process of finalizing Volume 7 of the Best of Public Domain Software which is to be about Chess. This letter is an invitation to any IACU chess players to contribute any programs, chess puzzles, insights on computer chess (e.g. weaknesses in the C-10 chess program under certain situations) or anything else.

If anyone has any material could they send it on a DSDD CDOS disk to:

Tom Beer
Applied Environmetrics
118 Gordon Street
Balwyn, VIC 3103
AUSTRALIA

All contributors will receive a copy of the final chess disk as well as the return of their own disk.

PLOT 10 Graphics Software

PLOT 10 Graphics Software for 4010 and 4110 series Tektronix terminals has been implemented to run using Cromix on a System 300.

PLOT 10 IGL is a library of Fortran '77 subroutines based on the SIGGRAPH Core proposal, developed for use in application programs requiring 2D and 3D graphics. Full control of color displays, line smoothing and contour algorithms, and high resolution fonts are available. Intelligent and simple terminals are supported, including emulation of many features for device independence.

This is not a product offering but a request to see if any interest for this type of product is outstanding. Pricing would start at $1,500.00 for the basic library. For more information, please contact:

Vories Pittman
CAS, Inc.
555 Sparkman Drive
Suite 1022
Huntsville, AL 35816-3424

MacroTech Memory Boards — Need help

I am writing hoping that one of the IACU members can help me with a "little" problem.

About one year ago, I bought a MacroTech 1MB memory board, (Max-M) with the memory map option, for one of my Cromemco System Three computers. The main idea was to use the extra memory as "RAM Disk." At the time of ordering, I was advised that the board could be configured to appear as a series of 64KZ boards.

Once I had the board, however, I was unable to configure the board appropriately. This was mainly due to the fact that I do not understand the configuration instructions. The configuration is done in software, but since the only language I speak a little of is "solder," I would only be able to come up with a hardware cure. I have written to both the manufacturer, who is not familiar with CDOS, and therefore is unable to help, and to suppliers, claiming in their advertisements that it can be done, to no avail. In all cases I have made my willingness

to pay for the information known.

The net result of my efforts is that I now have to run CP/M 2.2 in order to have the full use of the board, or be struck with a $1995 64K memory board in CDOS.

I would greatly appreciate it if you could put me in contact with anyone who uses the Macro-Tech memory boards in a Cromemco system, or wave a little flag in I/O NEWS to find someone.

Please address correspondence to:

C.W. Vlaun
MTCE/31, PMB 2418
Lagos, Nigeria
West Africa

The flag is waving. If anyone can help out, please forward the information to I/O NEWS as well (we don't know of anyone using the Macro-Tech memory boards).
Ed.

WriteMaster Project Underway

In the last two issues of *I/O NEWS*, an advertisement was run by **Quinn Team** soliciting pledges to finance the design of a version of WriteMaster tailored to the Cromix and UNIX environment. It was successful, and the project is currently underway at Quinn Team. The new version of WriteMaster will be re-written in 'C' to support the expanded capabilities of the 68000 environment. Quinn Team will continue to accept $100 pledges: contributing individuals will be able to participate in beta-testing, contribute suggestions as to what features they would like to see added, and of course, be the first to get a copy. Those interested should contact:

QUINN TEAM
30313 Canwood Street
Agoura, CA 91301
(818) 889-4819

A PCPU?

John Bush, of **Practical Peripheral Support** (aka Peripheral Labs), would like to know how many people would be interested in a processor board for Cromemco systems which would enable users to run MS-DOS/PC-DOS software. The design of the board, named the PCPU, has already been initiated. Whether or not it is ever manufactured will depend on the level of end-user response. So, if you would like to see such a board become available, contact John Bush at:

PPS
(aka Peripheral Labs)
1531 Kiowa Crest
Diamond Bar, CA 91765
(714) 861-6649 or (818) 339-5485

The Rumor Mill

► **68000 Diagnostic Software:** Word is spreading that Cromemco is nearing completion of a new suite of diagnostic software tools for 68000 systems. The new diagnostics are purported to be very user-friendly, with menus and interactive prompting. For example, the user is prompted for the system model, and from that, the software executes the appropriate tests based on the board components of that model. Although the diagnostics are currently being used in-house at Cromemco, the package won't be released to the public until the documentation has been completed — sometime later this year.

► **PRI Drivers for UNIX**: Rumor has it that the driver software for the PRI parallel printer interface will soon be available for UNIX systems. This would enable use of the PRI board thru which a parallel printer, such as the Cromemco 3355B fully formed character printer, could be interfaced. Presently, only serial printers can be used interfaced thru the Octart.

THE HACKER'S HOME

THE HACKER'S HOME explores techniques in 'C' programming for Cromix and UNIX. Users are encouraged to submit utility programs of their own. It is edited by Rick Dhaenens, Senior Manager for the Cromemco, Inc. Atlanta Regional Office, 5901-C Peachtree Dunwoody Rd, Suite 375, Atlanta, GA 30328. (404) 391-9433.

Programming Utilities for Fun

I have been sitting here for days trying to write this article. The biggest problem seems to be in defining what a useful program is. Is a program useful only for it's application result or for the programming concepts?

I have come to the conclusion that for the purposes of this column it should be both. This is hard to do because most of the utilities that are written are nothing fancy. Nobody cares about the code, only the results. It is very hard to conform to good programming practice when the object is to get the job done as fast as possible.

Programming is an art that we all practice with some degree of skill, but being human we tend to limit ourselves to a small subset of a given language. This is the subset of the most useful instructions for the objectives we set out to accomplish. To write a program to be used as an example one tries to fully use the features of the source language. In C as well as in other structured languages the emphasis is on proper structure and attention to data types. C in particular uses data structures and pointers extensively.

The operating environment must also be taken into consideration. System calls to set modes on IO channels are necessary for many interactive types of programming and some system calls are as useful as several lines of code in a higher level language. Environment utilities such as termcap are also useful but limit the portability of programs

I wrote the vdump utility for Unix because I did not like the OD (octal dump) utility. vdump was conceived as a regular hex dump program such as the Cdos or Cromix Dump utility but soon evolved into a visually oriented interactive dump program. Using vdump on Unix was so nice I decided to port it to Cromix as well. Most of the code was the same for both operating systems except for setting the terminal modes and the method finding the file size. The differences were small enough to be taken care of by IFDEF statements to include the appropriate routines at compile time.

The idea behind the final version of vdump (there were several) was that for any file position, calculations could be made to determine where to put the data on the screen. Everything could then be referenced directly from the desired byte in the file. The user interface was modeled on the Cromemco screen editor so most commands will be easy to use.

The workings of vdump are pretty straight forward. The specified file is opened, once we have opened the file the size of the file is obtained by the Cromix cstat or Unix fstat system calls. If the file name was not an ordinary file then the file size is set large enough for any device. Certain default parameters are set to their default values and calculations are made to determine the dump width and the screen limitations. The input modes are set to raw so a character at a time may be input, then the screen is cleared and the first screen of data is painted. vdump waits for any input from the keyboard and executes the appropriate function.

The most interesting routine is the *screen refresh* routine. This routine does all of the hard work, range checking and optimization of screen IO. It compares the requested file position with the beginning and ending file positions: if the request is valid it compares the requested and current positions to see if they are on the same (current) screen. If they are, the cursor is moved to the correct character. If the new position is within a half page up or down, the screen is scrolled to the appropriate position. Two special cases are then checked to see if the new position is on the first or last pages — the first screen always starts at the top left of the screen and the last page always ends on the bottom line of the screen. If none of these conditions are met then the line on which the new position resides is placed in the center of the screen.

There are several options that could be added to this utility to make it more useful. A query option would be very easy to implement as well as an option to substitute bytes into the file. This utility could also be used as a base on which to build other screen oriented interactive programs.

I hope you get as much use from this program as I have, and that the concepts presented will help with other programming applications.

```
/*****************************************************************************/
/*    VDUMP: a hex and ascii dump utility for Cromix and Unix.               */
/*    Syntax: VDUMP filename                                                 */
/*    Written by: Rick Dhaenens                                              */
/*    Date: 02/15/86                                                         */
/*****************************************************************************/


#include <stdio.h>                      /* Include the standard I/O file.   */

/********************     DEFINE OPERATING ENVIRONMENT     *********************/

#define CROMIX  1                       /* Define as CROMIX or UNIX system. */

/********************    INCLUDE FILES FOR SYSTEM CALLS    *********************/

#ifdef CROMIX                           /* Include these files for Cromix.  */
#include <modeequ.h>                    /* Equates for GETMODE & SETMODE.   */
#include <jsysequ.h>                    /* Equates for CSTAT system call.   */
int oml, om3;                           /* Storage for old tty modes.       */
#endif

#ifdef UNIX                             /* Include these files for Unix.    */
#include <termio.h>                     /* Equates for IOCTL system call.   */
#include <sys/types.h>                  /* Equates for Unix data types.     */
#include <sys/stat.h>                   /* Equates for STAT system call.    */
static struct termio ttold, ttraw;      /* Define terminal mode structures. */
static struct stat fstats;              /* Define STAT structure for file.  */
#endif

/********************          GLOBAL DECLARATIONS.          *********************/

long lseek();                           /* Forward decleration of function. */
extern int errno;                       /* External system error number.    */
static long size, curpos, topleft;      /* File size, current pos and top.  */
static int fd, dwidth, nlines, ncol;    /* File discriptor and dump width.  */
static int dir, rept, pagsiz;           /* Default direction, rept and etc. */

/********************           MISC DEFINITIONS.            *********************/

#define ERR     -1                      /* Define error value.              */
#define NCOL    80                      /* Display width in characters.     */
#define NLINES  24                      /* Display height in lines.         */

/********************        START OF MAIN PROGRAM           *********************/

main(argc, argv)                        /* Main for HEX and ASCII dump.     */
int argc;                               /* Arguement count.                 */
char *argv[];                           /* Pointers to arguements.          */
{                                       /* Beginning of MAIN.               */
  int c;                                /* Character variable.              */
  long addr;                            /* Address input variable.          */
```

```
/*******************        CHECK ARGS AND OPEN FILE        *********************/

  if (argc < 2) {                          /* Check if no arguements.          */
    printf("\n VDUMP:  Must have filename as arguement.\n");       /* Error.  */
    printf(" Syntax: VDUMP filename\n\n");       /* Print help message.      */
    exit(ERR);                             /* Exit with error if no args.      */
  }                                        /* End of IF.                       */

  if ((fd = open(argv[1], 0)) == ERR) {    /* Open the input file.             */
    printf("\n VDUMP:  ERROR %d Can't open file '%s'.\n\n",errno,argv[1]);
    exit(ERR);                             /* Exit with error if not found.    */
  }                                        /* End of IF.                       */

/*******************        GET THE SIZE OF THE FILE        *********************/

#ifdef CROMIX                              /* Use CSTAT for Cromix file size.  */
  if ((c = (cstat(fd, st_ftype) & 7)) == is_ordin)       /* Check if device.  */
    size = cstat(fd, st_size);             /* Qrdinary file so set size.       */
  else                                     /* Else.                            */
    size = 0xffffffff;                     /* Not file so set max size.        */
#endif

#ifdef UNIX                                /* Use FSTAT for Unix file size.    */
  fstat(fd, &fstats);                      /* Get file status.                 */
  if ((fstats.st_mode & S_IFMT) == S_IFREG)       /* Check if input is dev.    */
    size = fstats.st_size;                 /* Ordinary file so set size.       */
  else                                     /* Else.                            */
    size = 0xffffffff;                     /* Not file so set max size.        */
#endif

/*******************     SET DEFAULTS AND MISC VALUES      *********************/

  dir = 1;                                 /* Set direction to forward.        */
  rept = 0;                                /* Set default repeat to 0.         */
  curpos = 0;                              /* Set current position to 0.       */
  topleft = 0;                             /* Set top left screen position.    */
  ncol = NCOL;                             /* Set for terminal width.          */
  nlines = NLINES;                         /* Set for terminal height.         */

  dwidth = (((ncol-10)/4)-1);              /* Calculate dump width in bytes.   */
  if ((dwidth % 4) != 0)                   /* Is dump width an even 4.         */
    dwidth -= (dwidth % 4);                /* No then make it even.            */

  pagsiz = dwidth*nlines;                  /* Calculate characters per page.   */

/*******************          INITIALIZE SCREEN           *********************/

  setraw();                                /* Set RAW tty modes.               */
  cls();                                   /* Home the Cursor.                 */
  ref(-11);                                /* Refresh the entire screen.       */

/*******************     PROCESS KEYBOARD CHARACTERS       *********************/

  while ((c = getchar()) != 0x03) {        /* Get input from the keyboard.     */
    switch(c) {                            /* Switch on the incomming char.    */

      case  8:                             /* Ctrl H, move Cursor left.        */
        if (rept == 0) rept = 1;           /* Normalize repeat factor.         */
        ref(curpos-rept);                  /* Update screen.                   */
        break;

      case 10:                             /* Ctrl J, move Cursor down.        */
        if (rept == 0) rept = 1;           /* Normalize repeat factor.         */
        ref(curpos+(rept*dwidth));         /* Update screen.                   */
        break;

      case 11:                             /* Ctrl K, move Cursor up.          */
        if (rept == 0) rept = 1;           /* Normalize repeat factor.         */


        ref(curpos-(rept*dwidth));         /* Update screen.                   */
        break;

      case 12:                             /* Ctrl L, move Cursor right.       */
        if (rept == 0) rept = 1;           /* Normalize repeat factor.         */
        ref(curpos+rept);                  /* Update screen.                   */
        break;

      case 13:                             /* <CR>, move Cursor to left side.  */
        if (rept == 0) rept = 1;           /* Normalize repeat factor.         */
        ref(curpos-(curpos%dwidth)+(dwidth*rept*dir));  /* Update screen.    */
        break;

      case 'v':                            /* V, refresh screen;               */
      case 'V':                            /*                                  */
        ref(-1);                           /* Force screen refresh.            */
        break;

      case '+':                            /* + > or . Set direction forward.  */
      case '>':                            /*                                  */
      case '.':                            /*                                  */
        dir = 1;                           /* Set direction multiplier to 1.   */
        break;

      case '-':                            /* - < or , Set direction reverse.  */
      case '<':                            /*                                  */
      case ',':                            /*                                  */
        dir = -1;                          /* Set direction multiplier to -1.  */
        break;

      case '?':                            /* ?, The help function.            */
        cls();                             /* Clear the screen.                */
        setnorm();                         /* Set normal tty modes.            */
        printf ("\n\tCtrl C                  Abort program.\n");
        printf ("\tCtrl H  (Left arrow)   Move Cursor left.\n");
        printf ("\tCtrl J  (Down arrow)   Move Cursor down.\n");
        printf ("\tCtrl K  (Up arrow)     Move Cursor up.\n");
        printf ("\tCtrl L  (Right arrow)  Move Cursor right.\n");
        printf ("\t'+', '>', or '.'       Set direction to forward.\n");
        printf ("\t'-', '<', or ','       Set direction reverse.\n");
        printf ("\t'\$'                    Shell to operating system.\n");
        printf ("\t'0-9'                  Add to repeat factor.\n");
        printf ("\t'H'                    Set Cursor to HOME.\n");
        printf ("\t'JB'                   Jump to the beginning of file.\n");
        printf ("\t'JE'                   Jump to the end of file.\n");
        printf ("\t'JH<addr>'             Jump to hex address.\n");
        printf ("\t'JD<addr>'             Jump to decimal address.\n");
        printf ("\t'J.<addr>'             Jump to decimal address.\n");
        printf ("\t'JO<addr>'             Jump to octal address.\n");
        printf ("\t'P'                    Move repeat pages in direction.\n");
        printf ("\t'V'                    Refresh CRT screen.\n");
        printf ("\n\tPress any key to contunue.  ");     /* Prompt.           */
        setraw();                          /* Set RAW tty modes.               */
        c = getchar();                     /* Wait for key.                    */
        cls();                             /* Clear the screen.                */
        ref(curpos);                       /* Refresh data to screen.          */
        break;

      case '$':                            /* $, Shell to O.S.                 */
        cls();                             /* Clear the screen.                */


        setnorm();                         /* Set normal tty modes.            */
        system("sh");                      /* Shell to the O.S.                */
        setraw();                          /* Set RAW tty modes.               */
        cls();                             /* Clear the screen.                */
        ref(curpos);                       /* Refresh data to screen.          */
        break;

      case '0':                            /* Process numerical input.         */
      case '1':                            /* Used for the repeat function.    */
      case '2':                            /* Rept starts out as 0. Each time  */
      case '3':                            /* a key is pressed the value of    */
      case '4':                            /* rept is multiplied by 10 and the */
      case '5':                            /* value of the incomming number is */
      case '6':                            /* added. Routines that use repeat  */
      case '7':                            /* check if 0 (for 1 time) then use */
      case '8':                            /* the value then reset rept to 0.  */
      case '9':
        rept = (10*rept)+(c-0x30);         /* Bump up repeat factor.           */
        break;

      case 'h':                            /* H, Move Cursor to home position. */
      case 'H':                            /*                                  */
        ref(topleft);                      /* Refresh screen.                  */
        break;

      case 'j':                            /* J, Jump to beginning or end.     */
      case 'J':                            /*                                  */
        c = getchar();                     /* Get the next keyboard char.      */
        switch(c) {                        /* Switich on incomming character.  */

          case 'b':                        /* Jump to beginning if "b" or "B". */
          case 'B':                        /*                                  */
            ref(0);                        /* Refresh screen.                  */
            break;

          case 'e':                        /* Jump to end if "e" or "E".       */
          case 'E':                        /*                                  */
            if (size != 0xffffffff)        /* Check if valid end.              */
              ref(size-1);                 /* Refresh screen.                  */
            else
              beep();                      /* Beep at console operator.        */
            break;

          case 'd':                        /* Jump to absolute address.        */
          case 'D':                        /* Decode input in decimal.         */
          case '.':                        /*                                  */
            scanf("%ld", &addr);           /* Get address in decimal.          */
            goto dojmp;                    /* Calculate num, cx, and cy.       */

          case 'x':                        /* Jump to absolute address.        */
          case 'X':                        /* Decode input in hexadecimal.     */
            scanf("%lx", &addr);           /* Get address in hex.              */
            goto dojmp;                    /* Calculate num, cx, and cy.       */

          case 'o':                        /* Jump to absolute address.        */
          case 'O':                        /* Decode input in octal.           */
            scanf("%lo", &addr);           /* Get address in octal.            */
            goto dojmp;                    /* Calculate num, cx, and cy.       */

dojmp:      c = getchar();                 /* Strip off CR that stops scanf.   */

            ref(addr);                     /* Refresh the screen.              */
            break;

          default:                         /* Can't figure it out.             */
            beep();                        /* Then beep.                       */
        }                                  /* End of SWITCH.                   */
        break;

      case 'P':                            /* P, Move a page(s) in direction.  */
      case 'p':                            /*                                  */
        if (rept == 0) rept = 1;           /* Normalize repeat factor.         */
        ref(curpos+(dir*rept*pagsiz));
        break;

      default:                             /* If no matching command found.    */
        beep();                            /* Beep at console operator.        */
    }
  }

/*******************     CLOSE FILE AND EXIT TO O.S.       *********************/

  setnorm();                               /* Set normal tty modes.            */
  close(fd);                               /* Close input file.                */
  moveto(1,nlines);                        /* Go to bottom of screen.          */
  putchar('\n');                           /* Scroll everything up one.        */
  exit(0);                                 /* Return 0 for normal exit.        */
}

/*******************  OPTIMIZED SCREEN REFRESH ROUTINE   *********************/

ref(newpos)                                /* Refresh screen based on pos.     */
long newpos;
{
  int i, tmp;                              /* Local variables.                 */
  long newtop, pos;
  static int x, y = 1;

  if (newpos == -1)                        /* IF this is refresh only.         */
    pos = curpos;                          /* Set same position.               */
  else
    pos = newpos;                          /* Transfer to working variable.    */

  if (pos < 0 || pos >= size) {            /* IF out of range.                 */
    beep();                                /* Beep.                            */
  }
  else {                                   /* ELSE we have good numbers.       */
    if (pos >= topleft && pos < (topleft+pagsiz)) {  /* IF on same page?       */
      newtop = topleft;                    /* Don't refresh the screen.        */
      y = ((pos-topleft)/dwidth)+1;        /* Set line on same screen.         */
    }
    else if (pos > (topleft-pagsiz/2)  && pos <= topleft) {  /* ELSE IF up?   */
      for (i=0; i<(topleft-(pos-(pos%dwidth)))/dwidth; i++) {
        moveto(1,1);                       /* Go to top left of screen.        */
        insline();                         /* Insert blank line.               */
        dump(topleft-dwidth*(i+1));        /* Dump new line to the screen.     */
      }
      topleft -= dwidth*i;                 /* Update topleft.                  */
      newtop = topleft;                    /* Don't refresh the screen.        */
      y = 1;                               /* We are at the top.               */
    }

    else if (pos >= (topleft+pagsiz)  && pos < (topleft+1.5*pagsiz)) {  /*Dn?*/
      for (i=0; i<((pos-(topleft+pagsiz))/dwidth)+1; i++) {
        moveto(1,nlines);                  /* Go to bottom left of screen.     */
        printf("\n");                      /* Scroll everything up one.        */
        dump(topleft+pagsiz+dwidth*i);     /* Dump new line to the screen.     */
      }
      topleft += dwidth*i;                 /* Update topleft.                  */
      newtop = topleft;                    /* Don't refresh the screen.        */
      y = nlines;                          /* We are at the bottom.            */
    }
    else if ((pos/dwidth) < nlines) {      /* ELSE IF on first page?           */
      newtop = 0;                          /* Set addr 0 as home location.     */
      y = (pos/dwidth)+1;                  /* Start on first page.             */
```

Continued on page 33

NEW PRODUCTS...

NEW PRODUCTS is a regularly appearing column devoted to announcing and following hardware and software products of interest to Cromemco users. Most information is derived from press releases submitted by vendors. As a result, I/O NEWS cannot be responsible for errors of ommission or any other inaccuracies.

3270 BISYNC and ITI/X.25

Cromemco has released two new telecommunication software packages for their 68000 Cromix-Plus and UNIX systems. **3270 BISYNC** enables telecommunication between a Cromemco system and an IBM mainframe over a synchronous modem connected to a Cromemco Biart. **ITI/X.25** extends the same telecommunications capabilities to X.25 networks such as Telenet and Tymnet and also requires the Biart.

The price for each package is $795; orders can be placed with your local Cromemco dealer or directly from the factory. When ordering, specify the part number as follows:

3270 BSC-D-S/L: 3270 Bisync Communications Software for 68000 Cromix

ITI/X.25-D-S/L: X.25 Communications Network for 68000 Cromix

3270 BSC-X-S/L: 3270 Bisync Communications Software for UNIX

ITI/X.25-X-S/L: X.25 Communications Network for UNIX

Cromemco, Inc.
280 Bernardo Ave
P.O. Box 7400
Mountain View, CA 94039
(415) 964-7400

For a more detailed description of 3270 Bisync and X.25 protocols see the front cover article "New Telecommunications Software Links Cromemco to Mainframes."

Cromemco 68000 Structured Basic Compiler

Cromemco has announced the availability of its new **68000 Structured Basic Compiler** (model STBC-D (L/S)) for Cromix-Plus. Programs written using Cromemco's 68000 Structured Basic Interpreter (model STB-D) provide the source code from which machine executable code for the 68000 is generated by the Structured Basic Compiler (SBC). The compiler provides the speed advantages of a compiled language, while retaining all of the features of 68000 Structured Basic, including KSAM capabilities.

Because the compiler utilizes the SAVEd files of the 68000 Structured Basic Interpreter (SBI), which in turn can utilize programs written in Cromemco's Z80-based 32K Structured Basic (STB), the SBC provides an upward path from Z80 to 68000 performance levels for existing Structured Basic applications. In going from STB to STB-D, existing applications can be modified to take advantage of the better Cromix interface afforded by STB-D, which in turn can be passed to the compiled programs of the SBC.

The Structured Basic Compiler is available in March/April. The cost is $995. Orders can be placed through your local Cromemco Dealer or directly from Cromemco. When ordering specify Model STBC-D and disk size (Large or Small).

You can expect to hear more about this exciting product in upcoming issues of I/O NEWS.
Ed.

dBIII Compiler for Cromix-Plus

Software Standards has announced the availability of the Cromix-Plus port of WordTech's dBIII Compiler. This software will compile dBASE II and dBASE III command files into 68000 machine executable code with accompanying increases in speed.

It also provides a means for enhancing current dBASE II applications with the non-interactive features of dBASE III (10 open files as opposed to 2, date field support, etc) as well as providing a better interface to the Cromix-Plus operating system.

For additional information contact:

Software Standards, Inc.
6191 Choctaw
Baton Rouge, LA 70805
(504) 355-8024

See the front cover article, "dBIII Compiler for Cromemco Systems Under Cromix-Plus," for a more complete description of this product.

RDISK

RDISK adds up to 16 additional fixed disk drives to your IBM PC/XT/AT or compatible microcomputer system. You can have up to 1200 megabytes of disk storage without adding a fixed disk drive or controller. These disk drives are actually files on a remote computer system, but your PC treats them as local fixed disk drives. Any IBM PC software that can be run from a fixed disk can be run from an RDISK drive. Spreadsheet templates, text boiler plates, data bases, and data files can all be accessed from an RDISK drive.

To run RDISK you need:

► An IBM PC/AT/XT or compatible system with the following minimum configuration:
- 128 KB of memory
- A diskette drive
- A serial communication adapter or internal modem
- PC-DOS (or MS-DOS) 3.0 or higher
- RDISK/PC software

► A remote computer system with RDISK/REMOTE installed

► A means of establishing a session between the PC and the remote system

RDISK/REMOTE runs on a variety of computer systems:
- UNIX systems such as Cromemco 100/300/400, AT&T 3B2, AT&T 7300
- Minicomputer and Superminicomputers such as DEC VAX, DG Eagle, and IBM System 36
- Mainframe systems such as IBM 4300 and IBM 308x
- An IBM PC/XT/AT or compatible

In addition, DecTek/PC, a full VT100 emulator with file transfer and automatic script facility, can be purchased with the RDISK/PC at a special reduced price. This program can automate the remote system login and RDISK/REMOTE startup.

Prices:

RDISK/PC	$195
RDISK/REMOTE	starting at $495
DecTek/PC	$95

RDISK is available at your local Cromemco dealer or from:

Modular Systems
6425 Darlington Road
Pittsburg, PA 15217
(412) 521-6700
Dealer inquiries invited.

For more information on RDISK see the feature article "RDISK: Virtual DOS Disks Under UNIX"

68020 Piggyback Board

Computer System Associates, Inc. (CSA) has announced the availability of its **CSA 68020/6881 Piggyback Board**. The

CSA 68020/68881 Piggyback Board plugs directly into any 68000 or 68010 socket and perfectly emulates the existing 68000/68010 system, with a dramatic boost in performance. Since the 68020 is fully downward compatible with the 68000 or 68010, all existing software stays alive. In addition, the full power of the **68881 floating point math processor** chip is available for new software. Performance improvements of 500% or more can be expected in calculation intensive systems. Running the Berkeley Puzzle benchmark showed a 222% performance improvement over the 68000 at 12.5 MHz; the EDN linked list insertion Benchmark showed a 248% improvement.

The 68020 Piggyback Board will match all Motorola 68000/68010 parameters to 12.5 mhz in existing 68000 systems. For engineering laboratories designing 68020 equipment, the CSA 68020 piggyback board provides a quick, inexpensive method for prototyping a 68020/68881 system.

For upgrading the performance of an existing system operating system, the CSA 68020 piggyback board provides an easy, economical solution.

The 68020 Piggyback board with the 68881 math coprocessor sells for $1480.00. With just the 68020 on the piggyback board, it sell for $975.00. If you have access to the chips, the bare board is $575.00.

For additional information, or placement of orders contact:

Computer System Associates, Inc.
7564 Trade Street
San Diego, CA 92121
Contact Patricia Chouinard
(619) 566-3911
Telex 333693

I/O NEWS has arranged for Richard Quinn of Quinn Team to be provided with an evaluation Piggyback Board for the purpose of writing a review. You can expect to see Rich's review of the board in operation on Cromemco equipment in an upcoming issue (hopefully the next one).

Tango and Outbound

Tango, by **COSI**, is a communications package that allows a Personal Computer (PC) to share information with computers running the UNIX operating system. Tango is unique in its provision for emulation of multiple terminal types, file transfer in both directions, and host control over both file transfer and program execution on the PC.

Tango provides simple-to-use UNIX utilities for transferring files between the PC and the UNIX host and for invoking programs on the PC.

Features:

- Emulation of DEC VT-52, VT-100, IBM 3101, and Tektronix 4014 terminals.
- Transmission rates from 300 to 9600 baud.
- Support for Rixon, U.S. Robotics, Hayes Smartmodem, Novation Smartcat, and compatible modems.
- File transfer, ASCII or binary, in either direction with data conversion.
- Host control of PC file transfer and program execution.
- Modem and terminal command scripts

Tango can be ordered from your local Cromemco dealer or direct from Cromemco. There are two separate packages whose model numbers and prices are as follows:

TANGO-PC — PC to UNIX Communications Software (includes PC software only) $295.00.

TANGO-X — PC to UNIX Communications Software (includes UNIX software only) $445.

Outbound is an on-line communications program for UNIX users offering the same functionality as Tango. In addition, both Xmodem and Kermit error correcting file transfer protocols are provided. For more information regarding Tango and Outbound contact:

COSI
313 North First Street
Ann Arbor, MI 48103
(313) 665-8778
Telex 466568

Hacker's Home *Continued from page 31*

```
    }
    else if (pos > (size-pagsiz)) {      /* ELSE IF on last page?          */
      if(size%nlines == 0)               /* IF even number of characrers.  */
        newtop = (size-pagsiz)-dwidth;   /* Set home position.             */
      else
        newtop = (size-(size%dwidth)-pagsiz+dwidth); /* Set home position. */
      y = ((pos-newtop)/dwidth)+1;       /* Set close to bottom.           */
    }
    else {                               /* ELSE set it in the middle.     */
      newtop = pos-(pos%dwidth)-((nlines/2)*dwidth); /* Set home position. */
      y = (nlines/2)+1;                  /* Set active line to center.     */
    }
    if (newtop != topleft || newpos == -1) {  /* IF Refresh entire screen? */
      cls();                             /* Clear the screen.              */
      if (nlines < (size/dwidth)) {      /* IF the file is big enough.     */
        tmp = nlines;                    /* Fill whole screen.             */
      }
      else {                             /* ELSE find out how many lines.  */
        tmp = ((size%dwidth) == 0) ? (size/dwidth) : ((size/dwidth)+1);
      }
      for (i = 0; i < tmp; i++) {        /* Refresh for all lines.         */
        moveto(1,i+1);                   /* Move Cursor to correct line.   */
        dump(newtop+i*dwidth);           /* Dump line to the screen.       */
      }
      topleft = newtop;                  /* Update the home position.      */
    }
    if (pos != -1)                       /* IF not refresh only?           */
      curpos = pos;                      /* Update the current position.   */
    x = 10+((curpos%dwidth)*3)+((curpos%dwidth)/4); /* Find X Cursor pos.  */
    moveto(x,y);                         /* Move Cursor there.             */
  }
  rept = 0;                              /* Reset repeat factor.           */
}

/********************       DUMP ONE LINE TO SCREEN        *********************/

dump(byte)                               /* Dump block in hex and ASCII.   */
long byte;                               /* Byte is starting byte of block.*/
{                                        /* Beginning of DUMP.             */
  int i, len, cpos;                      /* Local variables.               */
  char buffer[512];                      /* Set up buffer for read.        */

  if ((lseek(fd, byte, 0)) == ERR)       /* Seek to byte in file.          */
    printf("\n VDUMP:  Seek error %d at byte %lx hex (%ld decimal).\n\n",
           errno, byte, byte);           /* Print seek error message.      */

  if ((len = read(fd, buffer, dwidth)) == ERR)      /* Read block from file. */


    printf("\n VDUMP:  Read error %d at byte %lx hex (%ld decimal).\n\n",
           errno, byte, byte);           /* Print read error message.      */

  printf("%06lx:  ", byte);              /* Print address of first byte.   */
  cpos = 10;                             /* Set current line position.     */

  for (i=0; i < len; i++) {              /* Process characters in buffer.  */
    printf("%02x ", (unsigned char)buffer[i]);  /* Print char in hex.       */
    cpos += 3;                           /* Advance current line position. */
    if (((i+1) % 4) == 0) {              /* Check if on boundary of 4.     */
      printf(" ");                       /* Yes then print extra space.    */
      cpos++;                            /* Increment Cursor position.     */
    }
  }
  if (len < dwidth) cpos--;              /* Compensate for column breaks.  */
  for (i=cpos; i < 9+(dwidth*3)+(dwidth/4); i++) {      /* Skip to ASCII.  */
    printf(" ");                         /* Pad with space character.      */
  }
  for (i=0; i <= len - 1; i++) {         /* Print ASCII representation.    */
    if (((char)buffer[i] <= 0x1f) || ((char)buffer[i] >= 0x7f))  /* Is Ctl? */
      printf(".");                       /* Print period if not printable. */
    else                                 /* Else.                          */
      printf("%c", buffer[i]);           /* Print the character.           */
  }                                      /* End of FOR.                    */
}                                        /* End of DUMP.                   */

/********************       SET THE INPUT MODES TO RAW      *********************/

setraw()
{
#ifdef CROMIX                            /* Use SETMODE for Cromix modes.  */
  om1 = setmode(STDIN,MD_MODE1,ECHO,ECHO);      /* Turn input echo off.   */
  om3 = setmode(STDIN,MD_MODE3,BINARY,BINARY);  /* Set binary mode.       */
#endif

#ifdef UNIX                              /* Use IOCTL for Unix modes.      */
  ioctl(0,TCGETA,&ttold);                /* Get copy of tty modes to save. */
  ioctl(0,TCGETA,&ttraw);                /* Get a copy to change.          */
  ttraw.c_lflag &= ~(ISIG|ICANON|ICRNL|ECHO);   /* Set raw mode no echo.   */
  ttraw.c_cc[4] = 1;                     /* Set min chars to 1.            */
  ttraw.c_cc[5] = 1;                     /* Set timeout to .1 second.      */
  ioctl(0,TCSETA,&ttraw);                /* Set terminal to raw modes.     */
#endif
}


/********************       RESTORE INPUT MODES        *********************/

setnorm()
{
#ifdef CROMIX                            /* Use SETMODE for Cromix modes.  */
  setmode(STDIN,MD_MODE3,om3,BINARY);    /* Restore binary mode.           */
  setmode(STDIN,MD_MODE1,om1,ECHO);      /* Restore echo mode.             */
#endif

#ifdef UNIX                              /* Use IOCTL to restore Unix modes.*/
  ioctl(0,TCSETA,&ttold);                /* Restore original tty modes.    */
#endif
}


/********************       ROUTINES TO CONTROL THE CRT       *********************/

#define ESC     0x1b                     /* Define terminal Esc character. */

beep()                                   /* Send Bell character to console.*/
   { putchar(0x07); }
cls()                                    /* Clear the CRT.                 */
   { putchar(ESC), putchar('E'); }
insline()                                /* Insert line on terminal.       */
   { putchar(ESC), putchar('L'); }
moveto(x,y) int x,y;                     /* Position the CRT Cursor.       */
   {putchar(ESC),putchar('F'),putchar(0x1f+y),putchar(0x1f+x); }
```

LOCAL USER GROUPS

Arizona Association of Cromemco Users
Contact: Jo Ann Drake, President
2207 West Eugie Avenue
Phoenix, AZ 85029
(602) 993-9589

Australia User's Group*
Contact: Minicomp
Minicomp Building
104 Mount Street
North Sydney, NSW 2060
Australia
(02) 957-6800
Meets Monthly
*Publishes "Minicomp/Cromemco" a monthly newsletter

Bay Area Cromemco Users & Programmers (BACUP)
Contact: Raymond Barglow or Alan Walworth
United Word & Data Processing
2345 Fulton Street
Berkeley, CA 94704
(415) 841-0708 or (415) 548-2692

Cromemcohorts
Contact: Dr. Brent Lowensohn
4747 Sunset Blvd.
Los Angeles, CA 90027
(213) 667-8972

Cromemco Users' Group of Australia*
Contact: Tony Stringer
52 Beechwood Avenue
Greystanes, 2145
*Publishes a magazine "CROME-SOMA"

Cromemco Users' Group Holland (CUGH)
Contact: Joop Kohler, Secretary
P.O. Box 120
2910 AC Kieuverkerk a/d IJssel
The Netherlands 01803-13300

Cromemco Users' Group*
Contact: Peter Norman
The University of Newcastle Upon Tyne
Department of Chemical Engineering
Merz Court, Claremont Road
Newcastle Upon Tyne NE1 7RU
England
Newcastle 28511, Ext. 3278
*Publishes Cromemco Users' Newsletter (CUG)

Cromemco Users' Group Ontario, Canada
Contact: Lloyd Parker
Hiram Walker Resources, Ltd.
Suite 600
1 First Cnandian Place
Toronto, Ontario
Canada M5X 1A9
(416) 864-3349

Cromemco Users of Orange County, California
Contact: Michael Peterson
Accountability Systems
700 South Tustin Avenue
Suite B
Orange, CA 92667
(714) 639-4570
Meets third Tuesday monthly

Insystems Pty. Ltd.*
Contact: Norman Rosenbaum
337 Moray Street
South Melbourne, Victoria
3205 Australia
(03) 690-2899, Telex: AA30458
*Publishes "Cromemco UPDATE" a bi-monthly newsletter

Illinois Users' Group
Contact: Jim Knowles
P.O. Box 631
Elgin, IL 60120
(312) 695-7775

Indonesian Cromemco Users' Group (ICUG)*
Contact: Zafir M.A. Pontoh
Computation Lab
Department of Regional & City Planning
Bandung Institute of Technology
10 Ganesha
Bandung, Indonesia
(022) 82051 ext. 360
*Publishes "BERKALA ICUG" a monthly newsletter

Microcomputer Users' Group
Contact: Noble Bright
P.O. Box 1
Cape May, NJ 08204
(609) 884-2222
(609) 429-3838
Meets fourth Wednesday monthly

Northwest Association of Cromemco Users (NWACU)
Contact: Jim Illman
403 S. Brandon
Seattle, WA 98108
(206) 763-2099

North San Diego County Users' Group
Contact: Charles Mackey
P.O. Box 397
Fallbrook, CA 92028
(619) 728-6116
Located 30 mi. east of Oceanside

Assisting in the formation of Local Cromemco User Groups is one of the services performed by The IACU. We will be happy to provide you with a list of our members in your area, and recommend other contacts to help you organize and maintain a Cromemco computer users group. Just call or write I/O NEWS.

North Texas Cromemco Commercial Users' Group
Contact: Jerrell Johnson
1131 Winterwood
Lewisville, TX 75067
(214) 221-1437
Or call Rocky Hall
at (214) 398-1595
Meets first Wednesday bi-monthly

NY, NY Users' Group
Contact: Charles Perrella
45F Route 303
Valley Cottage, NY 10989
(914) 268-5137

SaCromemco Users
Contact: Alan Whitman
Box 244
Rancho Cordova, CA 95670
(916) 635-6070

Silicon Valley Cromemco Users
Contact: Allan O'Neill
(415) 969-3854 or Emily Ott (415) 854-5818
meeting place provided by:
MCM Enterprises
215 Hamilton Avenue
Palo Alto, CA 94301
Meets fourth Tuesday monthly

W.A. Cromemco Users' Group
Contact: Rae Canning
c/o The W.A. School of Computing
2/294, Rokeby Road
Subiaco, Western Australia 6008

West Germany Users' Group
Contact: Glynnis Long
Tesco GmbH
P.O. Box 10
8714 Weisentheid
West Germany
09383-1237
Total fluency in English & German

Wisconsin Cromemco Users' Group
Contact: Bob Ungemach
6249 West Browndeer Road
Browndeer, WI 53223
(414) 355-1451

Become a Member of The I.A.C.U. and have I/O News Sent Directly to You.

I.A.C.U. Application for Membership

Name: _____

Title: _____

Company: _____

Mailing Address: _____

_____

City: _____ State: _____

Country: _____ Zip / Postal Code: _____

Phone: _____ (Area code) _____ (number) _____ (extension)

Telex:: _____

Membership Rates in the United States:
( ) 1 yr. = $42.00 ( ) 2 yr. = $76.00
( ) 3 yr. = $99.00
Membership Rates in Canada and Mexico:
( ) 1 yr. = $49.00 ( ) 2 yr. = $90.00
( ) 3 yr. = $125.00
Membership Rates in all other countries:
( ) 1 yr. = $60.00 ( ) 2 yr. = $104.00
( ) 3 yr. = $149.00

☐ I've included an additional $8.95 for my three-ring reference binder (including first class postage in U.S., Canada & Mexico. International purchasers, please add $4.00 [U.S.] extra for postage.)

Please start my Membership in the International Association of Cromemco Users right away.

☐ I have enclosed my ( ) Check ( ) Money Order in the amount of $_____ (U.S. Dollars only please) for a ( ) year membership.
International applicants: add $10.00 (U.S.) if you prefer to pay through your local bank.

☐ Now Available in the U.S. and most other countries on your VISA or Mastercard
(U.S. Dollars drawn on U.S. banks only **please.** Please Type or Print Clearly)
( ) VISA / Mastercard _____ (Expiration Date)
_____ (Your Full VISA or Mastercard Number)
Date _____ Signature _____
_____ (Name exactly as it appears on card)

Attention CROMIX Users
☐ Include $25 + $4.95 handling for "CROMIX, A User's Guide" by Leigh Thomas. (Calif. add 6%)

Mail Your Membership Application to:
The International Association of Cromemco Users
P.O. Box 17658, Irvine, CA 92713 U.S.A.

We're Interested!

We're Really Interested...in what you have to say. Especially about how you use your system...the problems encountered and the solutions effected...unusual uses or environments...and any practical applications you would be willing to share with fellow members. These can be short notes for departments like 'bits & bytes...' and 'Tec Tips,' or full feature articles.

Contact Bill Jaenicke at I/O News for editorial guidelines or assistance. We're interested in unleashing your literary talents.

BACK ISSUES

of I/O News are available

CHANGE OF ADDRESS

Membership # _____
Type or Print

Your Name _____

New Address

_____

_____

New area code & phone No. ( ) _____

Sign Here _____

Date new address in effect _____

Send address changes to I/O News, P.O. Box 17658, Irvine, CA 92713

The International Association of Cromemco Users

Software Resource Survey

— PURPOSE —

The purpose of this survey is to further expand the list of known software applications running on Cromemco systems. By taking the time to complete and return the questionaire you will be doing yourself and Cromemco users worldwide a tremendous favor. Everyone needs software, but finding the package you need can be terribly frustrating. This survey could eliminate that frustration. So please, help us help you by returning the completed questionaire at your earliest convenience.

Thank you!

— INSTRUCTIONS —

There are three (3) sections to this survey. Put your name and address information in the USER section. Use the SYSTEM section to describe the Cromemco system on which the software is run. If you have different systems running different operating systems and software, photocopy the survey sheet and submit one for each system.

The survey sheet has space for four (4) software package descriptions. If you have more than that, make some additional photocopies. For each package, provide the following information: the **product name**, a brief **description** of the software (spreadsheet, wordprocessor, DBMS, etc.); if you had **problems** in getting the package up and running, so indicate with comment; indicate whether or not you are satisfied with the package. Also, please provide the address and telephone information about the software **vendor** or manufacturer so that we can submit more detailed Product Fact Sheets to them. If the application was designed in-house, please make note of it and indicate if you're interested in having the application included in the **IACU Software Resource Guide**. If so, we'll send you some Product Fact Sheets.

When completed, please mail the survey sheets to:

The I.A.C.U.
P.O. Box 17658
Irvine, CA 92713

USER

Name _____ Member No._____

Address _____

_____

City _____ State _____ Zip _____ Phone _____

Country _____

SYSTEM

Model: _____

Memory: _____

Disks: _____

Operating System: ☐ CP/M ☐ CDOS ☐ Cromix

☐ Cromix-D ☐ Cromix-Plus ☐ UNIX ☐ Other_____

SOFTWARE

Product Name: _____

Description: _____

Problem? ☐ Yes ☐ No _____

Like it? ☐ Yes ☐ No _____

VENDOR INFO

_____

_____

_____

_____

Product Name: _____

Description: _____

Problem? ☐ Yes ☐ No _____

Like it? ☐ Yes ☐ No _____

_____

_____

_____

_____

Product Name: _____

Description: _____

Problem? ☐ Yes ☐ No _____

Like it? ☐ Yes ☐ No _____

_____

_____

_____

_____

Product Name: _____

Description: _____

Problem? ☐ Yes ☐ No _____

Like it? ☐ Yes ☐ No _____

_____

_____

_____

_____

BEST IN SALES AND SERVICE

We understand your desire to have a local Cromemco dealer in your area, and we believe we have a perfect solution for your problem at this time.

We are THE MOST QUALIFIED facility in the U.S. as we are the first recipient of Cromemco's new award "Number 1 U.S. Dealer". Again we have received awards for "Outstanding Service" as well as ranking in the "Top 25 Dealers". Based in the greater Dallas area, we sell and service Cromemco equipment in all regions of the U.S., and in several foreign countries.

Due to our central location and the convenient access to DFW Airport, equipment can be received from any location in the U.S. and returned to the customer within three working days (excluding transit time), if our expedited service is requested.

Service and support are the reasons that we are successful at what we do. Simply said, sales backed by support means success and that means **Computer Crossroads**.

SPECIALS OF THE MONTH

GET OFF THAT OLD Z-80

DON'T WATCH THE WORLD GO BY !!!! All that time you spent writing 8 bit **32K Structured Basic** programs has not gone to waste!! **compile'em** The new Cromemco Structured Basic Compiler converts those old basic programs to an executable machine code with speed increases of up to 1500%. Reserve your copy **NOW !! for only $995.00 postpaid.**
(requires cromix plus operating system)

GET UP TO DATE — WITH 31.05 !!!!!

now you can convert to 31.05 cromix operating system for under $4600.00 — this includes 2 MEGS OF MEMORY - DPU - MCU - OPERATING SYSTEM -- AND MORE!

ALL FOR ONLY $4595.00

NEW!

Computer Crossroads new conversion service... We will convert your Z-80 structured basic programs to 68000 executable machine code. This low cost service is designed for the person who does not have the time to convert their own programs.

A LIMITED NUMBER OF NEW CROMEMCO DEALERSHIPS ARE NOW AVAILABLE. CALL FOR THE NEW DEALER PACKAGE ... IT'S WORTH LOOKING INTO!

COMPUTER CROSSROADS of AMERICA, INC.

SOFTWARE
SERVICE
HARDWARE

Where It All Comes Together

We will meet or beat any bonafide sales quote on Cromemco equipment and will make available the best in support at no extra cost.

6 Terrace Shopping Center ● Richardson, Texas 75081

FOR SUPER SERVICE CALL (214)231-6108 Telex: 4991118

SYSTEMS ATLANTA, INC.

CROMIX SLAVE PROCESSORS

8MHZ Z80H 256K 2 Serial Ports

Two processes per slave

FAST - FAST - FAST - FAST - FAST

Eliminate degradation of system performance when running CP/M programs like dBASE and WORDSTAR in multi- user environments

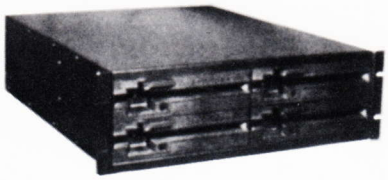
Runs under Cromix 11.27 and Cromix-Plus

$1095.00 including software license

Custom Disk Drive Cabinets and Power Supplies for Hard Disks, Cartridge Tapes, 5.25" and 8" floppies

Desk Top and Rack Mount

Call or write for information and prices



NEW 32K STRUCTURED BASIC

CROMIX (SBASIC.BIN)	$295
PC/MS DOS	$295
CP/M	$195
UPDATE CROMIX VERSION	$ 95 *
UPDATE CDOS VERSION	$ 75 *

* Available to registered Cromemco Licensees only.

FAST BACKUP for CROMIX Hard Disks $95
Backup to floppy disk at one megabyte per minute, 8" or 5.25"

FAST diskette copy programs $95
Copy both 8" and 5.25" CDOS, CROMIX and UNIX formats

Z8O cartridge tape backup utility $95
(Included with purchase of CTD tape subsystem at $1295)

CALL OR WRITE FOR INFORMATION ON ALL PRODUCTS AND SERVICES.

SYSTEMS ATLANTA, INC., P.O. BOX 99, LEBANON, GA 30146 (404) 928-0240

CROMEMCO COMPUTERS: DESIGNED TO MAKE UNIX SYSTEM V EVEN BETTER...

UNIX System V, the new standard in multi-user microcomputer operating systems, gives you high performance features along with the portability and flexibility of a standard.

Cromemco computers can make UNIX System V even better. Because our systems are designed with UNIX in mind. First of all, we offer UNIX System V with Berkeley enhancements. Then, our hardware uses advanced features like 64K of on-board cache memory and our high speed STDC controller to speed up disk operations – very important with UNIX.

More capability and expandability

We have a high-speed, 68000-based CPU that runs at 10 MHz, coupled with a memory manager that uses demand-paging and scatter loading to work *with* UNIX, not for it.

We provide room for expanding RAM to 16 megabytes – with error detection and correction – for running even the most sophisticated and advanced microcomputer programs. And the power to accommodate up to 16 users – all with plenty of memory.

But we give you even more.

A complete solution

We give you a choice in systems: the System 100 series, expandable up to 4 megabytes of RAM, and the System 300 series, expandable to 16 megabytes. A high speed 50 megabyte hard disk drive is standard on the systems. And you can expand the hard disk capacity up to 1200 megabytes using standard SMD drives. You can add floating point processing. High resolution graphics. Video digitizing and imaging. Communications through standard protocols. Mainframe interface.

And software support is here to meet your needs. We offer major programming languages, database management systems, communications software, including SNA architecture, X.25 protocol, and Ethernet; even a program to interface to an IBM PC if you need to. And, of course, access to the broad range of standard UNIX applications programs that is growing dramatically every day.

Easy to use.

We also make our systems easier to use, because we install the operating system before we ship your computer. No complicated installation procedures. And the Berkeley enhancements give you the standard UNIX System V operating system, but with the added convenience of these widely acclaimed improvements.

Cromemco's System 100 and System 300 computers: designed to be the highest performance UNIX systems available anywhere.

Just call or visit one of our UNIX System V Official System Centers to see for yourself. They'll also give you a copy of our new publication, "What you should know before you buy a UNIX system." Or contact us directly.

We'll be glad to show you how to get a better UNIX system.

Corporate Headquarters: Cromemco, Inc., 280 Bernardo Avenue, P.O. Box 7400, Mountain View, CA 94039. (415) 969-4710. In Europe: Cromemco GmbH, 6236 Eschborn 1, Frankfurter Str. 33-35, P.O. 5267, Frankfurt Main, Germany.

UNIX is a trademark of Bell Laboratories.
IBM is a trademark of International Business Machines Corp.

Cromemco®